

(19) 日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11) 特許出願公開番号
特開2003-169216
(P2003-169216A)

(43) 公開日 平成15年6月13日 (2003.6.13)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード* (参考)
H 0 4 N 1/41		H 0 4 N 1/41	B 5 C 0 5 9
H 0 3 M 7/30		H 0 3 M 7/30	A 5 C 0 7 8
H 0 4 N 7/30		H 0 4 N 7/133	Z 5 J 0 6 4

審査請求 未請求 請求項の数26 O L (全 24 頁)

(21) 出願番号	特願2001-364894 (P2001-364894)	(71) 出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22) 出願日	平成13年11月29日 (2001. 11. 29)	(72) 発明者	榎田 幸 東京都大田区下丸子3丁目30番2号 キヤ ノン株式会社内
		(72) 発明者	石川 智恵 東京都大田区下丸子3丁目30番2号 キヤ ノン株式会社内
		(74) 代理人	100076428 弁理士 大塚 康徳 (外3名)

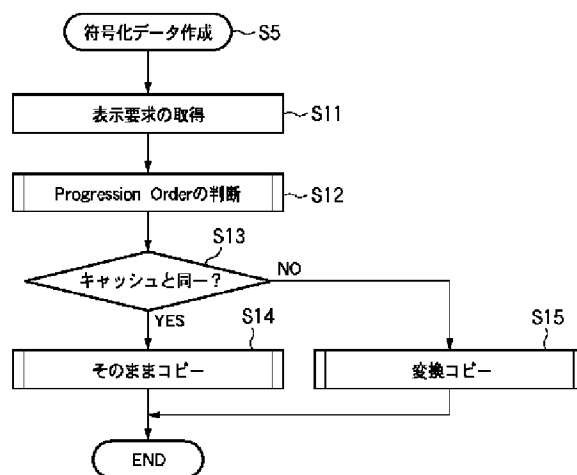
最終頁に続く

(54) 【発明の名称】 符号化データの作成方法及びその装置

(57) 【要約】

【課題】 断片化された符号化データを受信してキャッシュしておき、そのキャッシュした符号化データを用いて表示要求に応じた符号化データを作成する。

【解決手段】 断片化された符号化データを受信してメモリに格納しておき、ユーザの表示要求に基づいてProgression Orderを判断し (S 1 2)、その判断に基づいてヘッダ情報を変更し、その変更したヘッダ情報に基づいて、メモリに該当する符号化データが格納されているかどうかを判定し、メモリに格納されていると判定された断片化された符号化データを読み出して、その表示要求に適したProgression Orderに変換するとともに、メモリに格納されていない符号化データにはZ L Pを適用してJ P E G 2 0 0 0 準拠の符号化データを作成する。



【特許請求の範囲】

【請求項１】 断片化された符号化データを受信してメモリに格納する格納工程と、
ユーザの表示要求を取得する取得工程と、
前記取得工程で取得した前記表示要求に基づいてProgression Orderを判断する判断工程と、
前記判断工程での判断に基づいてヘッダ情報を変更する変更工程と、
前記変更工程で変更された前記ヘッダ情報に基づいて、前記メモリに該当する符号化データが格納されているかどうかを判定する判定工程と、
前記判定工程において前記メモリに格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記メモリに格納されていない符号化データにはダミー符号を適用してＪＰＥＧ２０００準拠の符号化データを作成する工程と、を有することを特徴とする符号化データの作成方法。

【請求項２】 前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データであり、前記符号化データはパケット単位に送信されることを特徴とする請求項１に記載の符号化データの作成方法。

【請求項３】 前記ダミー符号は、ＪＰＥＧ２０００で規定されているzerolength packetのデータであることを特徴とする請求項１に記載の符号化データの作成方法。

【請求項４】 前記変更工程では、画像全体のヘッダ情報を変更することを特徴とする請求項１に記載の符号化データの作成方法。

【請求項５】 前記変更工程では、タイル単位のヘッダ情報を変更することを特徴とする請求項１に記載の符号化データの作成方法。

【請求項６】 前記判断工程では、前記符号化データの解像度レベル及びレイヤ数と、前記表示要求の解像度レベル及びレイヤ数とに基づいて判断することを特徴とする請求項１に記載の符号化データの作成方法。

【請求項７】 断片化された符号データを受信してメモリに格納する格納工程と、
ユーザの表示要求を取得する取得工程と、
前記取得工程で取得した前記表示要求から表示に最低限必要とされる符号データを特定する特定工程と、
前記特定工程で特定された符号データを前記メモリから読み出し、その他の符号データをダミー符号に置き換えることにより、前記最低限必要とされる符号データを有する符号化データを作成する作成工程と、ことを特徴とする符号化データの作成方法。

【請求項８】 前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データでパケット単位に送信されることを特徴とする請求

項７に記載の符号化データの作成方法。

【請求項９】 前記ダミー符号は、ＪＰＥＧ２０００で規定されているzerolength packetのデータであることを特徴とする請求項８に記載の符号化データの作成方法。

【請求項１０】 前記最低限必要とされる符号化データの単位は、タイル単位であることを特徴とする請求項７に記載の符号化データの作成方法。

【請求項１１】 前記最低限必要とされる符号化データの単位は、パケット単位であることを特徴とする請求項７に記載の符号化データの作成方法。

【請求項１２】 前記特定工程では、前記メモリに格納されているか否かに拘わらず、前記表示要求に基づいて特定することを特徴とする請求項７に記載の符号化データの作成方法。

【請求項１３】 断片化された符号化データを受信して格納する格納手段と、
ユーザの表示要求を取得する取得手段と、
前記取得手段で取得した前記表示要求に基づいてProgression Orderを判断する判断手段と、
前記判断手段での判断に基づいてヘッダ情報を変更する変更手段と、
前記変更手段で変更された前記ヘッダ情報に基づいて、前記格納手段に該当する符号化データが格納されているかどうかを判定する判定手段と、
前記判定手段において前記格納手段に格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記格納手段に格納されていない符号化データにはダミー符号を適用してＪＰＥＧ２０００準拠の符号化データを作成する手段と、を有することを特徴とする符号化データの作成装置。

【請求項１４】 前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データであり、前記符号化データはパケット単位に送信されることを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１５】 前記ダミー符号は、ＪＰＥＧ２０００で規定されているzero length packetのデータであることを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１６】 前記変更手段は、画像全体のヘッダ情報を変更することを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１７】 前記変更手段は、タイル単位のヘッダ情報を変更することを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１８】 前記判断手段は、前記符号化データの解像度レベル及びレイヤ数と、前記表示要求の解像度レベル及びレイヤ数とに基づいて判断することを特徴とす

る請求項13に記載の符号化データの作成装置。

【請求項19】 断片化された符号化データを受信して格納する格納手段と、ユーザの表示要求を取得する取得手段と、前記取得手段で取得した前記表示要求から表示に最低限必要とされる符号データを特定する特定手段と、前記特定手段で特定された符号データを前記格納手段から読み出し、その他の符号データをダミー符号に置き換えることにより、前記最低限必要とされる符号データを有する符号化データを作成する作成手段と、ことを特徴とするデータの作成装置。

【請求項20】 前記断片化された符号化データは、ネットワークを介して送信されるJPEG2000符号化データでパケット単位に送信されることを特徴とする請求項19に記載の符号化データの作成装置。

【請求項21】 前記ダミー符号は、JPEG2000で規定されているzero length packetのデータであることを特徴とする請求項20に記載の符号化データの作成装置。

【請求項22】 前記最低限必要とされる符号化データの単位は、タイル単位であることを特徴とする請求項19に記載の符号化データの作成装置。

【請求項23】 前記最低限必要とされる符号化データの単位は、パケット単位であることを特徴とする請求項19に記載の符号化データの作成装置。

【請求項24】 前記特定手段では、前記格納手段に格納されているか否かに拘わらず前記表示要求に基づいて特定することを特徴とする請求項19に記載の符号化データの作成装置。

【請求項25】 請求項1乃至12のいずれか1項に記載の符号化データの作成方法を実行するプログラムを記憶したことを特徴とするコンピュータにより読取り可能な記憶媒体。

【請求項26】 請求項1乃至12のいずれか1項に記載の符号化データの作成方法を実行することを特徴とするプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、例えばネットワークを介して受信した断片的な画像データを、一つの画像ファイルに作成し直して符号化する符号化データの作成方法及びその装置に関するものである。

【0002】

【従来技術】インターネット上では、WWWサーバにWeb（ウェブ）ブラウザからアクセスして文書データや画像データ等を閲覧することが盛んに行われている。この仕組みはインターネット上に情報を公開するWWWサーバと、その情報を閲覧するクライアントを含み、各クライアントはウェブブラウザを使用して、そのサーバにより公開された情報を閲覧することができる。このWWWサー

バには、ホームページといわれる公開したい情報をHTMLで記述した文書があり、それをクライアント側のウェブブラウザがアクセスしてクライアントのコンピュータに表示する。またクライアント側のウェブブラウザは、表示しているページ内のリンクを辿っていくことにより、自分が必要な情報を得ることができる。更に、サーバが管理しているファイルをダウンロードする方法として、File Transfer Protocol（以下、FTPと略す）という方法がある。このFTPとは、ネットワークを通して、サーバ上にあるファイルを一度にクライアント・コンピュータに転送する仕組みである。

【0003】また画像ファイルへ断片的にアクセスして表示するためのプロトコルとしてFlashpix/IIPがある。このIIPは、Flashpixという、画像データファイルのフォーマットに最適なプロトコルになっており、画像データの部分アクセスの単位は、このFlashpixのタイル単位に行うものである。

【0004】一方、このIIPをそのままJPEG2000に適用した場合を考えると、JPEG2000では、各スケイラビリティの符号化データは、そのスケイラビリティより1つ下のスケイラビリティのデータからの差分データで構成されている。そのためクライアント側で受信した断片的な符号化データをキャッシュしておき、全符号化データをデコーダに引き渡して最初から再デコードする方法と、デコーダを途中で止めておいて今回受信した符号化データをデコーダに渡し、前回の続きからデコードする方法がある。

【0005】このような状況に鑑みて、マルチ解像度データを有する画像符号化データの中から必要な部分のデータのみを取り出して別の符号化データに変換する方法として従来から様々な方法が提案されており、例えば米国特許第6041143号（「MULTIRESOLUTION COMPRESSED IMAGE MANAGEMENT SYSTEM AND METHOD」）が挙げられる。この特許では、ソースとなる画像データは、ウェーブレット変換或いはウェーブレットのような変換を用いてマルチ解像度のデータを管理する符号化データである。そして、このソースとなる符号化データの中から、クライアントが選択した空間的な領域のデータを処理するために必要な符号化データを取り出し、1つの独立した符号化データに変換するものである。この時、ソースの符号化データから取り出される部分的な符号化データは、JPEG2000のコードブロックに対応しており、今回のクライアントからの要求を処理するために必要な符号化データを全ての階層で、かつ全てのサブバンドから該当する、或いは関係する全部の符号化データを含むものである。

【0006】

【発明が解決しようとする課題】しかし、符号化データをクライアント側で最初から再デコードする場合、サーバ側から送られてくる断片的な符号化データを、直接J

PEG 2000 準拠の 1 つの符号化データファイルに作り直すことも可能ではあるが、サーバ側から送られてくる断片的な符号化データは、クライアント側から要求する順に送られてくるため、この受信した符号化データを J P E G 2000 準拠の 1 本の符号化データに変換することは、クライアント側の処理が煩雑になり、コストがかかるという問題点がある。

【0007】又、サーバから送られてきた断片的な符号化データを独自形式のキャッシュファイルとして構成し、クライアント側の J P E G 2000 デコーダが直接このキャッシュファイルをリードすることも可能であるが、この場合、J P E G 2000 のデコーダは、この独自のキャッシュファイルをリードするための処理が必要となる。このため、処理が複雑になると共に、汎用の J P E G 2000 デコーダが使えないという問題点がある。

【0008】そこで、J P E G 2000 の画像をパケット単位で受信し、それらパケットデータをキャッシュしておく。そして、デコーダヘッダを渡す際に、まだ受信していないパケットデータ部分に、zero length packet (以下、Z L P と略す) データを挿入し、既に受信したパケットデータと合せて、一つのファイルを作成することで、メインヘッダの書き換えなど煩雑な処理を行わずに、一般的な J P E G 2000 デコーダで処理できる J P E G 2000 ファイルを作成する方法もある。

【0009】しかし、受信した全てのパケットデータを利用して作成された符号化データは、クライアント側のアプリケーションを使っているクライアントが要求した画像サイズ、画質および画像領域など、クライアントの望む表示形態を考慮していない。従って、この符号化データを受け取ったデコーダは、その一部のデータから生成されたデコード結果を利用して表示画面を作成することがあり、キャッシュされている全ての符号化データを使って 1 本の J P E G 2000 符号化データファイルを作るとは、デコーダ自体の処理にも無駄な処理が生じるという問題がある。

【0010】更に、キャッシュされている符号化データの量が多くなるほど、デコーダに渡す符号化データを作成するときに発生するデータのコピー作業が多くなる。これは、クライアントの表示に要するまでの時間が長くなることを意味しており、パフォーマンス低下という問題が生じる。特に、画像の拡大やスクロールという命令をクライアントが行うことで、表示に直接必要のないキャッシュデータが多くなり、上記の問題が頻繁に起こる。

【0011】また、クライアントが表示したい表示方法に最適な Progression Order は、クライアントの表示要求により変化する。以下、図 30 を用いて説明する。

【0012】図中、3001 は、解像度 (Resolution) が 4 個 (Resolution0~3)、レイヤ (Layer) が 3 個 (L

ayer0~2) ある場合を、Resolution Progression Order で格納した場合の符号化データを示す。一方、3002 は、符号化データ 3001 を SNR Progression Order で格納した場合のパケットの概念を示す。これら符号化データ及びパケットから、解像度が「0」、レイヤが「0」から「2」までをデコードする場合を考えてみる。符号化データ 3001 で示す Resolution Progression Order の場合は、3003a で示すように 3 個のパケットを連続して処理することでデコードできる。一方、パケット 3002 で示す SNR Progression Order の場合は、3003b で示す 3 個のパケットを処理するためには、3004 と 3005 で示す 6 個のパケットも処理しなければならない。しかし、これまでは、サーバ側が送りだした J P E G 2000 ファイルの Progression Order と、クライアントが表示したい表示方法に最適な Progression Order とが異なる場合でも、クライアント側で管理しているキャッシュファイルは、サーバ側で管理している J P E G 2000 のファイルの Progression Order と同じであるため、単純にキャッシュファイルから 1 本の J P E G 2000 符号化データに変換すると、クライアント側の J P E G 2000 デコーダに多大な負荷がかかるという問題点がある。

【0013】更に、上述した米国特許第 6041143 には、以下の問題点がある。

【0014】まず、この特許をネットワークを通した通信である I I P にそのまま適応した場合、クライアントからの要求の度に、サーバからクライアントに対して全ての階層の全コードブロックの符号化データを送ることになり、サーバはクライアントが既に受信済みの符号化データまでも送信することになる。更に、送信する符号化データのヘッダ部分は、今回要求した画像データの情報 (画像サイズ、階層情報など) に書き換えられており、元々サーバで管理している符号化データの情報を取ることができない。

【0015】本発明は上記従来例に鑑みてなされたもので、断片化された符号化データを受信して格納しておき、その格納している符号化データを用いて表示要求に応じた符号化データを作成することができる符号化データの作成方法及びその装置を提供することを目的とする。

【0016】又本発明の目的は、高速な符号化処理を可能にした符号化データの作成方法及びその装置を提供することにある。

【0017】

【課題を解決するための手段】上記目的を達成するために本発明の符号化データの作成方法は以下のような工程を備える。即ち、断片化された符号化データを受信してメモリに格納する格納工程と、ユーザの表示要求を取得する取得工程と、前記取得工程で取得した前記表示要求に基づいて Progression Order を判断する判断工程と、

前記判断工程での判断に基づいてヘッダ情報を変更する変更工程と、前記変更工程で変更された前記ヘッダ情報に基づいて、前記メモリに該当する符号化データが格納されているかどうかを判定する判定工程と、前記判定工程において前記メモリに格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記メモリに格納されていない符号化データにはダミー符号を適用してJ P E G 2 0 0 0 準拠の符号化データを作成する工程とを有することを特徴とする。

【0018】上記目的を達成するために本発明の符号化データの作成装置は以下のような構成を備える。即ち、断片化された符号化データを受信して格納する格納手段と、ユーザの表示要求を取得する取得手段と、前記取得手段で取得した前記表示要求に基づいてProgression Orderを判断する判断手段と、前記判断手段での判断に基づいてヘッダ情報を変更する変更手段と、前記変更手段で変更された前記ヘッダ情報に基づいて、前記格納手段に該当する符号化データが格納されているかどうかを判定する判定手段と、前記判定手段において前記格納手段に格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記格納手段に格納されていない符号化データにはダミー符号を適用してJ P E G 2 0 0 0 準拠の符号化データを作成する手段とを有することを特徴とする。

【0019】

【発明の実施形態】以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。

【0020】尚、本実施の形態は、ネットワークを介して受信した断片的な画像データを、一つの画像ファイルに作成し直す方法に関し、特にI S O / I E C - 1 5 4 4 4 に準拠したJ P E G 2 0 0 0 符号化データをインターネット・イメージング・プロトコル（I I P）などの画像符号化データの断片的な領域、或いは、断片的な符号化データを受信し、それらの断片化された符号化データを独自の方式でキャッシュするクライアントにおいて実施可能であり、更には、クライアントのアプリケーションプログラムにより要求される表示要求に応じて、クライアント側で受信した断片化された符号化データのキャッシュデータから1つのJ P E G 2 0 0 0 準拠の符号化データファイルを生成するものである。

【0021】図1は、インターネットに代表されるネットワーク上に複数のコンピュータが接続されている様子を示す概念図である。

【0022】図1において、100はインターネットに代表されるネットワークを示し、101、103はサーバコンピュータであり、それぞれ画像データを送信するためのJ P E G 2 0 0 0 用のI I Pサーバをはじめ、W W Wサーバ機能に必要なソフトウェアを実行しており、ま

たこれらサーバ101、103はそれぞれ大容量のハードディスク101a、103aを接続しており、そのハードディスクに大量の画像データも格納している。102a、102bはクライアントコンピュータであり、それぞれ、ウェブ（Web）ブラウザ、J P E G 2 0 0 0 デコーダなど、クライアント側で必要なソフトウェアを実行している。

【0023】図2は、サーバコンピュータ101、103或いはクライアントコンピュータ102a、102b等のハードウェア構成を示すブロック図である。

【0024】図2において、201はCPUで、装置全体の制御などを行っている。202はキーボードで、マウス202aとともに、この装置に情報などを各種データやコマンド等を入力するために使用される。203は表示部で、例えばCRTや液晶ディスプレイなどを備えている。204はROM、205はRAMで、この装置における記憶部を構成し、この装置が実行するプログラムや、この装置が利用するデータ等を記憶している。206はハードディスク装置、207はフロッピーディスク装置で、これらは、この装置で使用される外部記憶装置を構成している。208はプリンタである。209はネットワークを制御するネットワーク・インターフェースで、ここからインターネットなどのネットワーク100に接続されているサーバコンピュータなどのネットワーク上の資源にアクセスしたり（クライアントの場合）、クライアントからの要求に応じて符号化された画像や文書データを送信する（サーバの場合）ことができる。

【0025】本実施の形態では、既に生成済みのJ P E G 2 0 0 0 ファイルがサーバコンピュータ101、103で管理されている。一方、クライアントコンピュータ102a、102bでは、画像表示アプリケーションなど、クライアントが要求したJ P E G 2 0 0 0 ファイル内の必要な部分のみを、J P E G 2 0 0 0 用のI I Pを使用してサーバコンピュータ101、103に要求し、それに応答して送られてくる符号化データを断片的に受信し、この受信した断片的な符号化データをクライアントコンピュータ102a、102bの、例えばハードディスク206等にキャッシュする。このハードディスク206などにキャッシュされた断片的なJ P E G 2 0 0 0 符号化データからJ P E G 2 0 0 0 のデコーダに渡すための、J P E G 2 0 0 0 準拠の符号化データに変換する部分を説明する。

【0026】クライアント・コンピュータ102a、102bのユーザは、例えばウインドウズ（登録商標）

（Windows（登録商標））マシンを使用してホームページを開き、そこに書かれているJ P E G 2 0 0 0 の画像へのリンクをクリックすることにより、J P E G 2 0 0 0 の画像を、クライアント102a、102bの用途に適した画像サイズや解像度で表示するために必要

な断片的なデータとして取得しキャッシュする。そして、これらキャッシュデータから一本のビットストリームを作り出してデコードし、表示するという場合を想定する。

【0027】次に図3を参照して一般的なJ P E G 2 0 0 0の符号化データを説明する。

【0028】図3は、Layer-resolution level-component-position progression (以下、SNR Progressionと記す) に沿って記録されたJ P E G 2 0 0 0ファイルの構成を示す図である。

【0029】このSNR Progressionに準じた場合、(レイヤ) Layer / (解像度) Resolution / (成分) Component / (位置) Positionの順に記録される。このようなデータの並び方は、Progression Orderと呼ばれる。解像度(画像サイズ)とResolution番号との関係を図4に示す。

【0030】図4において、最も小さい解像度の画像の解像度(resolution)番号を「0」とし、resolution番号が一つ増加するごとに画像の幅と高さが共に倍になっていく。また各レイヤ(Layer)は、Resolution番号の小さい順に割り当てられている。レイヤ番号は、復元する画像の原画に対するS/N比に対応し、レイヤ番号が小さいほどS/N比が悪くなる。一つのJ P E G 2 0 0 0ファイル内でのResolution番号とレイヤ番号、component番号の最大値は、エンコードによって予め設定され、そのパラメータに従ってエンコードされており、その情報は符号化データの中に格納されている。また各パケット(packet)は、そのパケットに格納されているコードブロック(code-block)の情報を管理しているパケットヘッダ(packet header)部と、各コードブロックの符号化データから構成されている。

【0031】このようなJ P E G 2 0 0 0符号化データを使えば、クライアントはサーバ101(以下の説明では例としてサーバ101の場合で説明するが、サーバ103の場合も同様にして実行されることはもちろんである)にある全ての画像データを取得することなく、必要な部分のデータのみをサーバ101から受信することが可能である。クライアントの受信データの単位としては、J P E G 2 0 0 0のパケット(packet)、或はパケットよりも更に小さい符号化単位であるコードブロック単位が考えられる。本実施の形態では、クライアントがサーバから受信するデータ単位としてパケット単位を想定する。このパケット単位でのリクエスト及びレスポンスの概念図を図5に示す。

【0032】図5において、クライアント102aはサーバ101に画像のタイル番号(Tile)と(resolution level)とレイヤ(Layer)、component、position番号を指定してデータを要求する。図5の例では、タイル番号(Tile)は「1」、解像度「0」、レイヤ「0」、Componentは「0」、そしてPosition番号は「0」であ

る。これによりサーバ101は、画像503のコードストリームを解析して、その指定されたタイル番号、Resolution levelとLayer、component、position番号に相当するパケット(packet)データを抜き出し、パケット0としてクライアント102aに送り返す。

【0033】図6は、本実施の形態で使用するサーバ101で管理されているJ P E G 2 0 0 0ファイルの一例を示す図である。

【0034】図において、600は、SNR Progressionでエンコードされている例を示し、Layerは「0」～「2」の3種類、Resolution levelは「0」～「3」の4種類、componentとpositionは各1種類である。この場合の各パケットの順を601に示す。また説明を簡単にするために、サーバ101で管理しているJ P E G 2 0 0 0の最大解像度の画像サイズを1024×1024画素とする。よって4種類のResolution levelがあると、各解像度での画像サイズは、1024×1024、512×512、256×256、128×128画素となる。

【0035】図7は、クライアント102a(以下、代表して示す)のアプリケーションでの処理の概略を示すフローチャートである。ここでは、クライアント102aでJ P E G 2 0 0 0アプリケーションが起動しており、表示したいJ P E G 2 0 0 0ファイルが別の手段で指定された後での、画像の表示処理を示すフローチャートである。

【0036】まずステップS1で、ユーザが表示のための操作を行う。アプリケーションの起動時は、例えば開いたウィンドウサイズに収まる画像サイズを計算してもよい。次にステップS2に進み、ステップS1の操作により新たに必要となったパケットを全て算出する。その後ステップS3に進み、ステップS2の算出結果により新たに必要となったパケットデータをサーバ101に対して要求する。その後ステップS4に進み、その要求に応答してサーバ101から送られてくる、要求した全てのパケットデータを受信し、それをクライアントコンピュータ102aにキャッシュする。次にステップS5に進み、キャッシュ形式で管理している符号化データを、1本のJ P E G 2 0 0 0符号化データ形式に変換する。その後ステップS6に進み、ステップS5で作成したJ P E G 2 0 0 0符号化データをデコードして表示する。そしてステップS7に進み、ユーザが終了を要求した場合はアプリケーションを終了し、他の表示要求を行った場合は、再度ステップS1に戻って前述の処理を実行する。

【0037】[実施の形態1]次に、具体的な処理内容について説明する。

【0038】まずクライアント102aでのアプリケーションが起動した直後では、このアプリケーションの画面表示サイズは128×128画素であるとし、このサ

イズに入るサイズの画像データを表示する場合を考える。更にこの場合、図7のステップS2で算出される必要なパケットは、最低解像度の画像サイズで、SNRが最大のデータを表示することを考える。この場合は、図6における602の「Layer0/Reso0/Comp0/Pos0」、603の「Layer1/Reso0/Comp0/Pos0」、604の「Layer2/Reso0/Comp0/Pos0」で示す3個のパケットが必要となる。従ってサーバ101のプログラムは、これら3個のパケットを切り出す処理を行う。図6から分かるように、SNR Progressionで格納されていると、602と603の間、603と604の間には、今回のデコード処理では必要ないパケットが各3個ずつ入っている。

【0039】次に図7のステップS5の符号化データ生成の詳細な処理を図8乃至図11のフローチャートを参照して以下に詳しく説明する。

【0040】図8は、図7のステップS5における「符号化データの作成」処理を示すフローチャートである。

【0041】まずステップS11で、クライアントコンピュータ102aのユーザがアプリケーションに対してどのような表示要求を出したかを取得する。次にステップS12に進み、ステップS11で取得した表示要求から、JPEG2000のデコーダが高速に処理できるProgression Orderを判断する。この判断処理は、図9のフローチャートを参照して後述する。次にステップS13に進み、ステップS12で判断したProgression Orderと、キャッシュされているProgression Orderとが同一か否かを判断する。同一の場合はステップS14に進み、不一致の場合はステップS15にそれぞれ移動する。ステップS14ではProgression Orderが同一であるため、キャッシュ内の形式通りにコピーしてJPEG2000符号化データファイルを生成する。一方、ステップS15では、Progression Orderを変換しながら、JPEG2000符号化データファイルを生成する。

【0042】次に図9を参照して、図8のステップS12における「Progression Orderの判断」処理について説明する。

【0043】まずステップS21、ステップS22にて、今回、サーバ101に要求したJPEG2000符号化データファイルの解像度レベル(resolution level)数、レイヤ(Layer)数を各変数Ro、Loに取得する。次にステップS23、ステップS24では、ステップS11で取得した、ユーザが今回指定した表示要求に基づいて、表示用の解像度レベル数、レイヤ数を、それぞれ変数Rq、Lqに取得する。尚、これら変数Ro、Lo、Rq、Lqは、上述したRAM205に設定されている。その後ステップS25に進み、各変数の値に基づいて $Rq \times Lo$ 、 $Lq \times Ro$ をそれぞれ計算し、その計算結果を比較する。ここで $Rq \times Lo < Lq \times Ro$ の場合はステップS26に進み、それ以外の場合はステップS27に進み、夫々「Resolution Progression」、「SNR Progression」と判断する。

【0044】具体例を使って説明する。ステップS21とステップS22で、それぞれ $Ro=4$ (解像度0~3)、 $Lo=3$ (レイヤ0~2)、ステップS23とステップS24で各々 $Rq=1$ (128×128 であるため)、 $Lq=3$ (レイヤ0~2)となる。この結果、 $Rq \times Lo=3$ 、 $Lq \times Ro=12$ となって $Rq \times Lo < Lq \times Ro$ を満足する。これによりステップS26に進み、「Resolution Progression」が選択される。これにより図8のステップS13の判断において、異なるProgression Orderであると判断されてステップS15の変換コピー処理に移動する。

【0045】図10は図8のステップS14の「そのままコピー」の詳細処理を示すフローチャート、図11は図8のステップS15の「変換コピー」の処理の詳細を示すフローチャートである。

【0046】まず図10を参照して、「そのままコピー」の処理について説明する。

【0047】まずステップS31、S32で、タイル数とパケット数を取得し、それぞれ変数iTMax、iPMaxに記憶する。次にステップS33に進み、JPEG2000のメインヘッダを出力する。次にステップS34に進み、内部変数iTを「0」で初期化する。尚、これら変数は、前述したようにRAM205に記憶される。

【0048】次にステップS35に進み、タイルヘッダ(Tile Header: SOTマーカコード)を生成し、仮出力する。次にステップS36、S37で、パケット数をカウントする変数iP、パケットデータのバイト数を計数する変数iLを共に「0」に初期化する。次にステップS38に進み、変数iPで示されるパケットのデータがキャッシュにあるか否かを判断する。キャッシュにある、即ち、サーバ101から転送済みであればステップS39に進み、キャッシュに存在しなければステップS42にそれぞれ移行する。ステップS39では、変数iPで示されるパケットデータを出力し、次にステップS40で、その出力したパケットデータのバイト数を変数iLに足し込む。一方、キャッシュにない場合はステップS42に進んでZLP(zero length packet)のデータを出力し、次にステップS43に進み、ZLPのバイト数、即ち、「1」を変数iLに足し込む。

【0049】こうしてステップS40或はステップS43を実行した後ステップS41に進み、変数iPをインクリメントし(+1)、次にステップS44に進んで、変数iPと変数iPMaxとを比較する。このステップS44は、タイル内の全てのパケット数分の処理を行ったかを判断するものである。ここで全てのパケットを処理していなければ再度ステップS37の処理に戻って前述の処理を実行するが、全てのパケットの処理が終わっていればステップS45に進み、タイル全体のデータ数を管理しているフィールドの値を変数iLから更新する。

そして、出力したSOTマーカコード内のタイル全体のデータ数を出力する。次にステップS46に進み、タイル数を計数する変数iTをインクリメント(+1)する。そしてステップS47に進み、変数iTの値が最大タイル数に到達したか、即ち、全てのタイルの処理が終わったか否かを判断する。終わっていないならばステップS35に戻って前述の処理を実行するが、全てのタイルの処理が終わっていればステップS48に進み、JPEG2000の符号化データを完成させるための処理を行う。ここでは、EOCマーカコードなどを出力する。

【0050】図11は、Progression Orderを変更しながらJPEG2000符号化データを生成する図8のステップS15の「変換コピー処理」を示すフローチャートで、この図11では、図10で示す処理と同じ処理を行う部分には同一の番号を付けている。よって、図10で示した処理と異なる部分のみを説明する。

【0051】ステップS31、S32で、全てのタイル数とパケット数とをそれぞれ変数iTMax、iPMaxに格納した後、ステップS50では、図8のステップS12で判断したJPEG2000のデコーダが高速に処理できるProgression Orderを示すメインヘッダのデータを生成して出力する。具体的には、図6の600で示すCODマーカコード内の「Progression Order」のフィールド値を変更する。図11において、点線500で囲んである部分の処理が図10で示したフローチャートと異なっている。即ち、ステップS51では、変数iPで示される順に出力すべきパケットを、ステップS50で変更した「Progression Order」から算出する。その後ステップS52では、実際にそのパケットデータを取得して出力する。残りの部分の処理は図10で説明したものと同一であるため、その説明を割愛する。

【0052】次に、本実施の形態に係る処理の具体的な例について説明する。

【0053】本実施の形態1の場合は、「Resolution Progression」が出力符号化データのProgression Orderとして選択されており、一方、キャッシュ内のProgression Orderは、「SNR Progression」であるために、図11で示す「変換コピー」処理が選択される。

【0054】ステップS31、S32では、タイル数は「1」であるため変数iTMax=1とし、図6に示すように12個のパケットがあるため、変数iPMax=12をセットする。次にステップS50では、図12の1105で示すように、CODマーカコード内のProgression Orderフィールドを「SNR Progression」を示す値「1」に変更してメインヘッダを書き込む。次にステップS34に進み、変数iTを「0」に初期化し、ステップS35で、タイルヘッダ「Tile Header」を仮に出力する。その後ステップS36、S37で、各変数iP、iTILELとともに「0」で初期化する。次にステップS51で、変数iPが指す順番に出力するパケットをキ

ャッシュから探す。最初は変数iPの値は「0」なので、1番目に出力するパケット、即ち、「Layer0/Reso0/Comp0/Pos0」のパケットデータを探す。次にステップS38に進み、このパケットデータがキャッシュ内にあるかを判断する。今回は既にキャッシュにあるためステップS52に進み、実際に「Layer0/Reso0/Comp0/Pos0」のパケットデータを取得して出力する。その後ステップS40に進み、その出力したパケットのバイト数を変数iTILELに足し込む。その後ステップS41に進み、パケット数をカウントする変数iPをインクリメント(+1)し、続くステップS44で、変数iPと変数iPMaxとを比較する。今回は $iP < iPMax$ でない、即ち、全てのパケットを読込んでいないためステップS37に戻る。そして変数iPの値が「1」、「2」の時は上記と同様の処理を行い、変数iPの値が「3」～「11」までは、ステップS38の判断によりキャッシュに存在しないと判断されるためステップS38からステップS42に進み、ZLPのデータを出力し、ステップS43で、変数iTILELをインクリメントする。こうしてステップS41の処理後、変数iPの値が「12」になるとステップS44の判断により $iP < iPMax$ でなくなってステップS45に進み、変数iTILELからタイルのバイト数を計算し、SOTマーカ内の該当する位置を変更する。次にステップS46に進み、変数iTの値をインクリメントし、ステップS47の判断により、全てのタイルを処理したかを判断する。本実施の形態では、タイルの数は1つなので、そのままステップS48に進んで終了処理を行なう。

【0055】この様にして図12に示すJPEG2000符号化データが完成する。

【0056】他の場合として、解像度を最大まで上げ、SNR最低の画像を表示する場合を考える。具体的には、解像度レベル「3」、レイヤ「1」までを表示する。この場合、ステップS3（図7）で算出される必要なパケットは、

「Layer0/Reso0/Comp0/Pos0」, 「Layer0/Resol/Comp0/Pos0」, 「Layer0/Reso2/Comp0/Pos0」, 「Layer0/Reso3/Comp0/Pos0」, 「Layer1/Reso0/Comp0/Pos0」, 「Layer1/Resol/Comp0/Pos0」, 「Layer1/Reso2/Comp0/Pos0」, 「Layer1/Reso3/Comp0/Pos0」,

の8個のパケットが実際のデコード処理に必要な。この場合のステップS5の符号化データの作成処理（図7）を説明する。

【0057】この場合は図9のステップS21とステップS22では、それぞれ $Ro=4$, $Lo=3$ となり、ステップS23、ステップS24で、各々 $Rq=4$, $Lq=2$ となる。従って、ステップS25で $Rq \times Lo=12$, $Lq \times Ro=8$ となるため、 $Rq \times Lo < Lq \times Ro$ が満足されずにステップS27に進み、「SNR Progression」が選

扱われる。これによりステップS 1 3の判断で、キャッシュと同一のProgression Orderであると判断され、ステップS 1 4の「そのままコピー処理」に移行する。その後、図1 0のフローチャートで示す処理を行い、図1 3で示すようなJ P E G 2 0 0 0の符号化データを生成する。

【0 0 5 8】これにより、クライアント1 0 2 aからいかなる表示要求が入力されても、デコーダが実際に処理するパケットデータが、できるだけ符号化データの先頭から連続する符号化データとするべく変換でき、デコード処理を高速に行うことができる。

【0 0 5 9】〔実施の形態2〕前述の実施の形態1では、メインヘッダ内のProgression Orderを変更する場合で説明したが、本実施の形態2では、表示に必要なタイルのみを変更する場合で説明する。

【0 0 6 0】図1 4（A）は、1枚の画像データを16個のタイルに分割した時の図を示す図である。

【0 0 6 1】図1 4（B）は、サーバ1 0 1で管理されている符号化データの例を示す図である。この符号化データは、前述の実施の形態1と同様に、「SNR Progression」で全タイルのデータが格納されている。本実施の形態2では、まず画像全体を表示するために、全タイルの解像度レベル「0」、レイヤ「0」の各パケットデータが送られて表示された後、タイル5（Tile 5）、6（Tile 6）、9（Tile 9）、10（Tile 10）の4個のタイルの部分だけ解像度を上げて表示する場合で説明する。この時に要求する解像度レベルは「1」とする。またこの条件でエンコードされ、サーバ1 0 1に格納されているJ P E G 2 0 0 0符号化データファイルの例を図1 4（B）の1 4 0 0に示す。この図に示すように、全てのタイルの符号化条件は同一であり、Progression Orderを示すものは、1 4 0 1に示す様に、メインヘッダ内のCODマーカに記載されている。

【0 0 6 2】このファイルに対してまず、画像全体を表示するための符号化データ、即ち、全タイルの解像度レベル「0」、レイヤ「2」を要求した後、図1 4（A）のタイル5、6、9、10のみを解像度を上げて表示することを考える。この場合、タイル5、6、9、10の解像度レベル「1」、レイヤ「2」のみを表示する場合を図1 5以降を参照して説明する。

【0 0 6 3】図1 5は、本発明の実施の形態2に係る符号化データの作成処理を示すフローチャートである。

【0 0 6 4】まずステップS 6 1で、表示要求をクライアント側のアプリケーションプログラムから取得するが、その時に表示するタイルの情報までも取得することが、前述の実施の形態1と異なっている。次にステップS 6 2に進み、前述のステップS 1 2と同様にして、ステップS 6 1で取得した表示要求から、J P E G 2 0 0 0のデコーダが高速に処理できるProgression Orderを判断する。次にステップS 6 3～S 6 5で、前述のステ

ップS 3 2～S 3 4と同様にして、タイル数を取得して変数i TMaxに記憶し、J P E G 2 0 0 0のメインヘッダを出力し、内部変数i Tを「0」で初期化する。次にステップS 6 6に進み、タイルデータを出力し、ステップS 6 7で、タイル数をカウントする変数i Tを+1し、ステップS 6 8で全てのタイルに対する処理が終了したかを調べ、終了していなければステップS 6 6に戻り、終了するとステップS 6 9で終了処理を実行して、この処理を終了する。

【0 0 6 5】図1 6は、図1 5のステップS 6 6のタイル出力処理を示すフローチャートである。

【0 0 6 6】図において、まずステップS 7 1では、タイルのヘッダの内、SOTマーカのみを出力する。このときSOTマーカのバイト数はこの時点では決まっていないので仮に出力する。これは前述の実施の形態1と同じである。次にステップS 7 2に進み、パケット数を計数する変数i Pを「0」にし、ステップS 7 3で、タイルのヘッダ部の処理を行う。この処理は図1 7のフローチャートを参照して後述する。

【0 0 6 7】次にステップS 7 4に進み、ステップS 7 3で設定された変数Sの値に応じて3種類の処理を切り替える。ここではステップS 7 4で、S=0であればステップS 7 6に進んで、全てのパケットデータをZLPに置き換える。ステップS 7 4でS≠0であればステップS 7 5に進み、S=1かどうかをみる。S=1、即ち、メインヘッダで出力したProgression Orderと本タイルのProgression Orderとが一致するときはステップS 8 1に進み、それがキャッシュされているかどうかを調べ、キャッシュされていればステップS 8 2に進み、キャッシュされている各パケットデータをキャッシュで管理されている順に出力する。またステップS 8 1でキャッシュされていない場合はステップS 7 6に進み、ZLPを出力する。

【0 0 6 8】一方、ステップS 7 5でS=1でない場合、即ち、Progression Orderが不一致の場合はステップS 8 3に進み、キャッシュされているかどうか調べ、キャッシュされている場合はステップS 8 4に進み、該当するパケットデータをキャッシュから取得して出力する。またステップS 8 3でキャッシュされていない場合はステップS 7 6に進み、ZLPを出力する。その他の処理は、前述の実施の形態1と同じである。

【0 0 6 9】次に、図1 6のステップS 7 3のヘッダ部の処理について、図1 7のフローチャートを参照して説明する。

【0 0 7 0】まずステップS 9 1で、タイル数を計数する変数i TileLに初期値「0」をセットする。その後ステップS 9 2に進み、今回処理しているタイルが、図1 5のステップS 6 1で取得した、デコードに必要なタイルか否かを判断する。このステップで今回処理しているタイルがデコードに必要な場合はステップS 9 3に

進む。

【0071】一方、必要なタイルの場合はステップS95に進み、Progression Orderが同じであるかどうかを判断する。この判断は、図15のステップS62で判断したProgression Orderが、図15のステップS64で処理したメインヘッダのProgression Orderと同一かどうかを判断する。同一の場合はステップS96に進み、異なる場合はステップS98に進み、CODマーカを出力する。このCODマーカの中には、このタイルのProgression Orderを記入するフィールドがあり、このフィールドに、図15のステップS62で判断したProgression Orderを示す値を書き込む。その後ステップS99に進んでSODマーカを出力する。尚、このフローチャートにおいて、ステップS93、S96、S99の処理は全て同一の処理である。その後、各ステップS94、S97、S100において、それぞれの状態に応じて変数Sの値を決定する。

【0072】図18は、実際に図14の符号化データ1400に対して、本実施の形態2で示す処理を行った場合に出力される符号化データの一例を示す図である。

【0073】図において、1800は、出力した符号化データ全体を示し、1801はメインヘッダ内のCODマーカコードの状態を示し、1801で示す、このメインヘッダ内部のProgression Orderは、サーバ101にあるファイルと同一の値である。またタイル5、6、9、10以外のタイルデータは、1802に示した値を持っている。即ち、今回のデコード及び表示には関係ないタイルとして、全てZLPのデータになっている。また、タイル5、6、9、10のデータは、先ず1803で示す様にタイルヘッダ内にCODマーカコードを出力し、更に、CODマーカコード内のProgression Orderを示す部分には、「Resolution Progression」を示す値がセットされている。これは、今回の表示要求から $Rq \times Lo = 2 \times 3 = 6$ 、 $Lq \times Ro = 3 \times 4 = 12$ となり、図9のステップS25にて $Rq \times Lo < Lq \times Ro$ と判定されてステップS26に進み、「Resolution Progression」と判断されるからである。1804は、タイル5の実際のデータを示している。

【0074】これにより、Progression Orderの変更が必要なタイルのみ変更処理を行い、更に、高速に変換処理を行うことができる。尚、このデコード処理は、前述の実施の形態1と同様に高速に行える。

【0075】以上説明したように、本実施の形態2に係るファイル作成方法を用いれば、IIPを用いて送られてくるJPEG2000符号化データの断片的なデータをキャッシュし、このキャッシュファイルから、通常のJPEG2000デコーダで処理できるJPEG2000符号化データを作る時に、今回のデコード及び表示に最適なProgression OrderでJPEG2000符号化データを再構築でき、デコード処理における余分なデータ

処理を省くことができ、デコード処理を高速に行うことができる。

【0076】また、デコード及び表示する領域をタイル単位で指示する場合は、そのデコード、表示に必要なタイルのみに対して処理を行うことができるので処理を高速化できる。更には、タイル毎に表示方法を変えるような複雑な表示方法に対しても対応することができる。

【0077】[実施の形態3]次に本発明の実施の形態3について説明する。尚、この実施の形態3においても、そのハードウェア構成は、前述の実施の形態と同様である。また、この実施の形態3においても、ユーザの要求する表示に必要なデータを、このパケット単位で受信し、表示する場合で説明する。

【0078】[タイル単位で書き込みデータを選択する方法]図19は、本発明の実施の形態3に係る処理を説明するフローチャートである。

【0079】まずステップS111において、クライアント102aのユーザは、画像全体が表示領域に納まる解像度のデータをサーバ101に要求する。例えば、オリジナル画像(abc.jp2)がサーバ101にあるとする。この画像(abc.jp2)は、最大解像度でサイズ1024×1024(画素)、8×8の64タイルに分割されており、コンポーネント(component)数が「3」で、解像度レベル(resolution level)が「2」、即ち、3つの画像サイズ方向の階層を持っており、2つのレイヤ(Layer)に分けられていると仮定する。それぞれのタイルには図20(A)に示すように、左上のタイルから順にシーケンス番号が付されている。この時、サーバ101のJPEG2000ビットストリームの構成は図21の2101に示すような形になる。

【0080】サーバ101のJPEG2000ビットストリームが、解像度(resolution)スケラビリティに準じるデータの並びになっているとすると、それぞれのタイルの中は、図21の2102に示すようなパケットの並びとなる。更に、クライアント102aにおける表示領域を256×256(画素)と仮定すると、この場合、クライアント102aはステップS111において、画像「abc.jp2」のヘッダデータ2103と、解像度(resolution)「0」を構成する全タイルのパケット、つまり、各タイルの中から図21の2104で表されるパケットデータを要求する。即ち、2(レイヤ)×3(成分)×64(タイル)×1(解像度)=384(パケット)を要求する。次にステップS112に進み、受信したパケットデータをキャッシュする。即ち、先ほど要求したヘッダデータと384(パケット)データをキャッシュする。次にステップS113に進み、キャッシュデータの中から、既に受信されていたヘッダデータ2101とパケットデータとを利用して、ステップS111で要求された表示画像に応じたJPEG2000準拠の符号化データを作成する。次にステップS11

4に進み、ステップS113で作成された符号化データをデコードして表示する。

【0081】次にステップS115に進み、ユーザは画像の興味のある一部分を最大解像度まで拡大する要求を出す。例えば、解像度レベル (resolution level) 2の中心部の領域2002 (図20 (B)) を表示するために、不足しているデータを要求する。つまりJPEG2000では、各解像度が差分データからできているので、タイル27, 28, 35, 36の解像度レベル (resolution level) 1と、解像度レベル (resolution level) 2を構成するパケットデータを要求する。即ち、タイル27, 28, 35, 36のデータの中のパケットデータ2105, 2106 (図21) で表される2 (レイヤ) \times 3 (成分) \times 4 (タイル) \times 2 (解像度) = 48 (パケット) を要求することとなる。次にステップS116に進み、不足分のデータを受信してキャッシュする。つまり、ステップS115で要求した48パケットのデータを受信してキャッシュする。

【0082】そしてステップS117に進み、既にステップS112, S116で受信してキャッシュしているデータから、ステップS115で要求された表示画像に応じたJPEG2000準拠の符号化データを作成する。次にステップS118に進み、ステップS117で作成されたデータをデコードして表示する。つまり、図20 (A) の解像度レベル (resolution level) 2の領域2001が表示される。

【0083】これらステップS113、ステップS117における符号化データ作成の動作を図22のフローチャートを参照して説明する。

【0084】まずステップS121では、クライアントの要求している表示形態を知るために、表示に必要なタイルの番号を把握する。例えば、ステップS116の場合には、ステップS115での表示要求から、表示に必要なタイルは27, 28, 35, 36であることを取得する。そしてステップS122では、サーバ101にあるオリジナル画像のタイル数Tmax、および各タイルを構成するパケット数Pmaxを取得する。これは、既にサーバ101から受信したJPEG2000のメインヘッダを解析して求めても良いし、サーバ101に問い合わせ求めても良い。

【0085】次にステップS123に進み、表示用ファイルつまり、デコーダへ渡す符号化データを書き出すファイルに、サーバ101から受信したメインヘッダをそのまま書き込む。次にステップS124では、タイル番号を計数するための変数カウンタTに「0」を入れて初期化する。次にステップS125では、タイルTのタイルヘッダを表示用ファイルに書き出す。ここでタイル長は、このタイルに含まれるパケットデータのバイト数に左右される。そのため、この時点では、正確なタイル長を知ることはできない。そこで、暫定的な値として、全

て「0」のzerolength packet (ZLP) である場合のSOTマーカを書き込む。このタイルのデータ長は、(タイルヘッダの長さ) + (タイルに含まれるパケット数) \times 1 (バイト) として計算する。例えば、タイルヘッダに含まれるマーカがSOTマーカのみで、1つのタイルに18パケットが含まれている場合は、ZLPは長さが1バイトであるので、タイルのデータ長は14 + 1 \times 18 = 32 (バイト) となる。

【0086】図23 (A) (B) は、タイル番号が「0」、含まれるパケット数が「18」であるタイルに対して作成されるSOTマーカの構成例を示す図である。

【0087】ステップS126では、カウンタTに示されるタイル番号をもつタイルが、ステップS121で取得した、表示に必要なタイル番号に含まれているか否かを判断する。含まれていなければ、つまり、表示に必要なタイルであればS138に進み、含まれていれば、つまり、表示に必要なタイルであればステップS127に進む。例えば、ステップS121で表示に必要なタイル番号として、27, 28, 35, 36の番号を取得していれば、これら4つの値とカウンタTの値を比較し、同じであればステップS127へ、違っていればステップS138へ進む。

【0088】ステップS138では、タイルTは表示に必要なタイルであると判断されているので、タイルTに含まれる全てのパケットデータの部分に、ダミーの値として、ZLPであることを示す1バイトの値「0」を書き出す。例えば、1タイルに含まれるパケット数Pmaxの値が「18」であれば、図24の2402で示すように、1バイトの値「0」を18バイト分、タイルヘッダ (SOTマーカ) 2107の後ろに書き込む。

【0089】ステップS127では、パケット番号のカウンタPに「0」を入れて初期化し、更に、タイルTにおいて表示に必要なパケットデータのデータ長をカウントするための変数Lengthに「0」を代入して初期化する。次にステップS128に進み、パケットPが既に受信されキャッシュされているかどうかを判断する。未受信のパケットデータであればステップS129に進み、そのパケットデータがキャッシュされていればステップS130に進む。例えば、タイル27に関しては、ステップS113においては、解像度レベル (resolution level) 0のパケットデータ2104 (図21) しか受信していないので、P=0~5に関しては、ステップS130へ、P=6~17に関してはステップS129に進むが、ステップS116において、最大解像度、最大SNRで、全てのコンポーネントのパケットデータを受信してキャッシュしているため、P=0~17の18パケットの場合は全てステップS130へ進むことになる。

【0090】ステップS129では、未受信パケットデータの処理であるので、パケットPのデータとしてZL

Pの値、即ち、1バイトの値「0」を表示用ファイルに書き込む。またステップS128でキャッシュされている場合はステップS130に進み、パケットPのデータをキャッシュから取り出し、そのまま表示用ファイルに書き込む。そしてステップS131に進み、ステップS129或はS130でファイルに書き込んだバイト数を変数Lengthに足す。例えば、ステップS129において、ZLPの値を書き込んだ場合には、変数Lengthには「1」が足され、ステップS130において、ファイルに書き込んだパケットPのバイト数が100バイトなら、変数Lengthには「100」が足される。そしてステップS132に進み、パケット数Pのカウント値を「1」だけ増やす。そしてステップS133で、値PがPmaxと同じ値かどうか判断する。同じ値であればステップS134に進み、同じ値でなければステップS128に戻る。

【0091】例えば、解像度レベル (resolution level) 数が「3」、レイヤ (Layer) 数が「2」、component数が「3」のタイルであれば、タイルは18パケットを含んでいるので、パケット数Pの値が「18」であればステップS134に進む。そうでなければステップS128に戻る。ステップS134では、変数Lengthの値からタイルTのタイル長を計算し直し、SOTマーカのタイル長を表すP_{sot}2301 (図23 (A)) の値を上書きする。例えば、タイルヘッダにSOTマーカのみが含まれ、変数Lengthの値が「286」であった場合には、このP_{sot}2301には、値 (14+Length=)

「300」が上書きされる。即ち、ステップS113で作成される表示用ファイルのタイル27, 28, 35, 36のデータ部分に関しては、サーバ101のファイルと全く同じ18パケットのデータ2102 (図21) が書き込まれることになり、図24で示されるようなファイルが作成される。

【0092】例えば、ステップS113において作成される表示用ファイルであれば、各タイルは表示に必要であり、2104で示される6個のパケットのデータしか受信していないので、表示用ファイルにおける各タイルのデータ部分には、サーバ101から受信したパケットデータ2104の後ろ、図21の2105, 2106で示す部分に、12バイトの「0」が書き込まれた (2505, 2506) 図25に示すファイルが作成される。

【0093】ステップS135では、タイル番号のカウントTを1だけ増やす。そしてステップS136では、全てのタイルに関して、上記のことが行われたかどうかを判断する。計数値TがTmaxと同じであれば、即ち、全てのタイルに関して処理が終了しているためステップS137へ進み、Tmaxと異なればステップS125に戻る。例えば、タイル数が「64」であれば、カウンタTの値が「64」であればステップS137に進み、カウンタTの値が「64」でなければステップS125に

進む。ステップS137では、JPEG2000のEOCコードである2バイトの値「0xFFD9」を書き込み、ファイルを閉じる。

【0094】このような符号化データの作成処理を採用することにより、表示に必要なタイルかどうかを判断するだけで、そのタイルのパケットデータのみが書き込まれ、その他のタイルのデータには、ZLPの値が書き込まれたファイルが簡単に作成できる。しかも、必要なタイルの未受信データ部分にもZLPを利用することで、簡単にJPEG2000準拠の符号化データファイルを作成できる。

【0095】[実施の形態4] 前述の実施の形態3では、タイル単位で表示に必要なデータであるかを判断し、必要であれば、そのタイルに関して受信したデータの全てを表示用ファイルに書き込み、そのタイル内の未受信のパケットに対しては、ダミーのデータとしてZLPを書き込んでいた。そして、表示に必要なではないタイルのデータは全てZLPを書き込んでいた。つまり、表示されるタイルに関しては、受信した全てのデータを書き込んでいたので、表示の際に必要なとされないパケットデータも含まれる可能性がある。従って、ユーザの表示要求に応じたファイルの作成という点ではまだ不十分である。

【0096】例えば、地図などのように、川崎市全体の地図を表示し、その後、「中原区今井上町53」というような住所で、その目的地周辺の詳しい場所を、最も解像度の高い画像で把握した後、画面を引いて最寄駅である「武蔵小杉駅」と目的地との位置関係を把握するような場合、前述の実施の形態3に係る方法では、表示に必要な無いパケットデータも含まれてしまう。従って、より表示要求と密接になった符号化データファイルを作成するためには、タイル単位でのデータの取捨選択に加えて、表示に必要なタイル内においてパケット単位でのデータの取捨選択を行う必要がある。

【0097】[パケット単位で書き込みデータを判断する方法] このように、一度、画像の一部分を最大解像度まで拡大した後、解像度を下げて表示範囲を広げるような動作は図26に示すステップS149～S152で表される。尚、この図26のフローチャートにおいて、ステップS141～S148の処理は、図19のフローチャートにおけるステップS111～S118の処理と同様であるため、その説明を省略する。

【0098】ステップS149では、ステップS148で表示した解像度よりも低い解像度で画像を表示し、表示領域に納まる画像の部分領域を広くするような要求を出す。例えば、ステップS148において、256×256 (画素) の表示領域に、図20 (A) に示す領域2001を、解像度レベル2 (resolution level 2) で表示していたとすると、ステップS149では、タイル18, 21, 42, 45で囲まれる領域を解像度レベル1

(resolution level 1) で表示する、つまり領域 2002 (図 20 (B)) を表示するように、ステップ S149 において要求を出す。次にステップ S150 に進み、ステップ S149 で要求された表示に必要なパケットで、キャッシュされていないデータを受信してキャッシュする。つまり、タイル番号 18~21, 26, 29, 34, 37, 42~45 の 12 個のタイルに含まれる解像度レベル (resolution level) 1 を構成するパケットデータ 2105 を受信してキャッシュする。従って、ステップ S150 では、2 (レイヤ) × 3 (成分) × 12 (タイル) × 1 (解像度) = 72 (パケット) を受信してキャッシュする。次にステップ S151 に進み、キャッシュデータから、表示に必要なパケットのみを使って符号化データを作成する。そしてステップ S152 に進み、ステップ S151 で作成された符号化データをデコードして表示する。尚、ステップ S151 では、キャッシュデータから必要なパケットデータのみから符号化データを作成する。

【0099】図 27 は、図 26 のステップ S151 の符号化データ作成時の処理動作を示すフローチャートである。尚、この図 27 におけるステップ S161~S167, ステップ S169 乃至 S178 の処理は、図 22 のステップ S121 乃至 S127, ステップ S129 乃至 S138 の処理と同じであるため、その説明を省略する。

【0100】まずステップ S161 では、クライアントの表示要求として、タイル番号の他に解像度レベル (resolution level)、レイヤ (Layer)、コンポーネント (component) を取得する。前述の前述の実施の形態 3 では、タイル番号のみでファイルにデータを書き込むか否かを判断していたが、本実施の形態 4 では、表示に必要なタイルの中から、更に表示に必要なパケットデータのみを表示用ファイルに書き込む。そのため、解像度レベル、レイヤ、成分といった表示の細かい条件が必要となる。そしてステップ S162 以降の処理で、前述の実施の形態 3 と異なるのは、ステップ S168 の処理であり、その他のステップの動作は、実施の形態 3 と同じであるので、その説明を省略する。

【0101】ステップ S168 では、表示に必要なタイルに含まれるパケットに関して、パケット P が要求された表示画面を作成するのに必要かどうかを判断する。必要なパケットデータであればステップ S170 に進み、不必要なパケットデータであればステップ S169 に進む。例えば、タイル 18 について、図 26 のステップ S150 において、解像度レベル 0, 1 を構成するパケットデータしかキャッシュされていないので、タイル 18 の場合には、キャッシュされているパケットデータは、全てステップ S168 を経てステップ S170 に進んで表示用ファイルに書き込まれる。しかし、タイル 27 に関しては、キャッシュされた全てのデータを表示画面の作

成時に利用するわけではない。解像度レベル 0 又は 1 を作成するパケットデータ P=0~11 に関しては、ステップ S168 において必要であると判断されてステップ S170 に進むが、解像度レベル 2 に相当するパケットデータ (P=12~17) は、キャッシュデータに存在するが、今回の表示には利用されないで、ステップ S169 に進み、そのキャッシュされているパケットデータの代わりに ZLP の値、即ち、1 バイトの値「0」がファイルに書き込まれる。従って、タイル 27, 28, 35, 36 に関しては、図 28 に示すようなデータ 2801 が表示用ファイルに書き込まれる。

【0102】図 28 は、キャッシュされているタイル 27 のパケットデータから表示用のデータ 2801 を作成するための処理を説明する概念図である。

【0103】ここでは、解像度 0, 1 で、レイヤが 1, 2、及びコンポーネント 0~2 であるタイル 27 のパケットデータがキャッシュされており、これがそのまま表示用ファイル 2801 として書き込まれている。また、解像度 2 以上のデータは表示には使用されないで、ZLP の値がその後に書き込まれている。

【0104】図 29 は、表示用ファイルにパケットデータが書き込まれる様子を説明する図である。

【0105】図 29 において、タイル 18 に関しては、解像度レベル 0, 1 を構成するパケットデータしかキャッシュされていないので、このタイル 18 の場合には、キャッシュされているパケットデータは、全て表示用ファイルに書き込まれる。

【0106】一方、タイル 27 に関しては、キャッシュされた全てのデータを表示画面の作成時に利用するわけではないので、解像度レベル 0 又は 1 を作成するパケットデータ P=0~11 に関しては必要であると判断されて表示用ファイルに書き込まれるが、解像度レベル 2 に相当するパケットデータ P=12~17 は表示には利用されないで、キャッシュされているパケットデータの代わりに ZLP の値、即ち、1 バイトの値「0」が書き込まれることになる。

【0107】このような符号化データの作成方法を採用することにより、表示に必要なタイルかどうかを判断するだけでなく、更に、そのタイルの中の必要なパケットデータのみが書き込まれ、その他のパケットデータには ZLP の値が書き込まれた JPEG 2000 準拠の符号化データファイルが簡単に作成できる。

【0108】以上の説明において、データのキャッシュ方法については、それぞれのパケットデータが必要とときに取り出せる形式であれば、それぞれのパケットを独立に管理しても、一つのファイルの中にまとめて保存されていても、その他どのような方法でも構わない。

【0109】以上説明したように、本実施の形態に係るファイル作成方法を用いることにより、IIP を用いて送られてくる JPEG 2000 符号化データの断片的な

データをキャッシュし、このキャッシュしたファイルから、通常のJPEG2000デコーダで処理できるJPEG2000符号化データを作る時に、今回、デコード／表示する部分のみの符号化データをコピーした必要十分な表示用ファイルを作成することが可能である。

【0110】これにより、1本の符号化データを作る処理において、データのコピー作業を削減でき、高速なファイル作成が可能になるという利点がある。

【0111】また、表示画面を作成するために符号化データをデコードする際に、デコード／表示に必要なないパケットデータ部分にダミーの値としてZLPのデータを利用することで、特別なデコーダを必要とせずに表示処理ができるという利点がある。

【0112】更に、表示画面を作成するために、デコーダ側で受け取った符号化データを取捨選択する必要性がなくなるという利点がある。また、作成するJPEG2000の符号化データファイルのサイズを小さくすることもできる。更に、JPEG2000のデコーダも、余計な部分を処理することがなく、高速に処理することもできる。

【0113】また本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダー、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0114】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても達成される。

【0115】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0116】プログラムコードを供給するための記憶媒体としては、例えば、フロッピーディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモ리카ード、ROMなどを用いることができる。

【0117】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0118】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わる

メモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0119】

【発明の効果】以上説明したように本発明によれば、断片化された符号化データを受信して格納しておき、その格納している符号化データを用いて表示要求に応じた符号化データを作成することができる。

【0120】又本発明によれば、高速な符号化処理を可能にできるという効果がある。

【図面の簡単な説明】

【図1】本発明の実施の形態に係るサーバ、クライアントを含むネットワークシステムの構成を示す概念図である。

【図2】本発明の実施の形態1に係るサーバ、クライアントの構成を示すブロック図である。

【図3】本実施の形態に係るJPEG2000符号化データの概念図である。

【図4】JPEG2000の解像度スケーラビリティを説明する図である。

【図5】本実施の形態に係るサーバ、クライアント間でのIIPのprotocolsの概念図である。

【図6】本実施の形態に係るサーバに格納されている符号化データの一例を示す図である。

【図7】本実施の形態に係るクライアントでの処理を説明するためのフローチャートである。

【図8】図7のステップS5の符号化データ作成処理を説明するフローチャートである。

【図9】図8のステップS12のProgression Orderの判断処理を説明するフローチャートである。

【図10】図8のステップS14の「そのままコピー」処理を説明するフローチャートである。

【図11】図8のステップS15の「変換コピー」処理を説明するフローチャートである。

【図12】本発明の実施の形態1に係る処理結果の一例を示す図である。

【図13】本発明の実施の形態1のその他の場合での処理結果の一例を示す図である。

【図14】本発明の実施の形態2に係るタイル構成の一例を説明する図である。

【図15】本発明の実施の形態2に係る符号化データ作成処理を示すフローチャートである。

【図16】図15のステップS66のタイル出力処理を示すフローチャートである。

【図17】図16のステップS73のヘッダ出力処理を示すフローチャートである。

【図18】本発明の実施の形態2における処理結果の一例を示す図である。

【図 1 9】本発明の実施の形態 3 に係るクライアントの画像要求動作を示すフローチャートである。

【図 2 0】J P E G 2 0 0 0 の解像度スケーラビリティとタイル分割の一例を示す図である。

【図 2 1】J P E G 2 0 0 0 のパケット単位による符号化データの構成例を示す図である。

【図 2 2】実施の形態 3 に係る表示用ファイルの符号化データ作成処理を示すフローチャートである。

【図 2 3】S O T マーカの構成を説明する図である。

【図 2 4】作成された表示用ファイルにおける非表示部分のタイルの符号化データの状態を説明する図である。

【図 2 5】解像度レベル (resolution level) 0 を表示

する際の表示用ファイルにおける表示タイルの符号化データの状態を説明する図である。

【図 2 6】本実施の形態に係るクライアントの画像要求動作を示すフローチャートである。

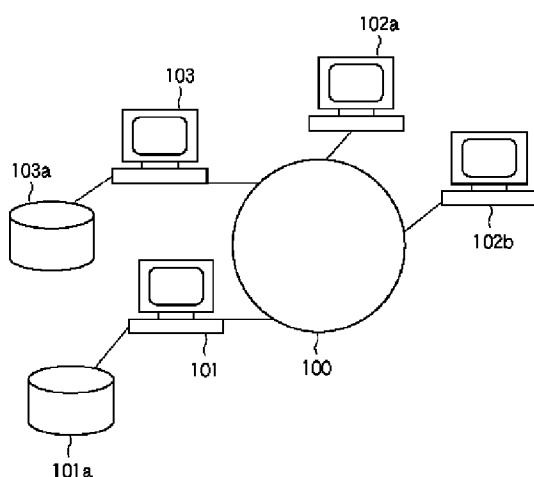
【図 2 7】実施の形態 3 における表示用ファイルの符号化データ作成処理を示すフローチャートである。

【図 2 8】実施の形態 4 におけるキャッシュデータと表示用ファイルデータを説明する図である。

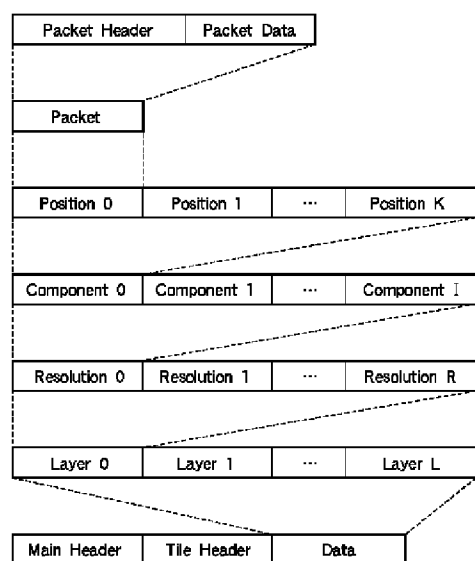
【図 2 9】本発明の実施の形態 4 に係る表示用ファイルデータ作成の様子を説明する図である。

【図 3 0】従来技術を説明するための J P E G 2 0 0 0 符号化データの構成図である。

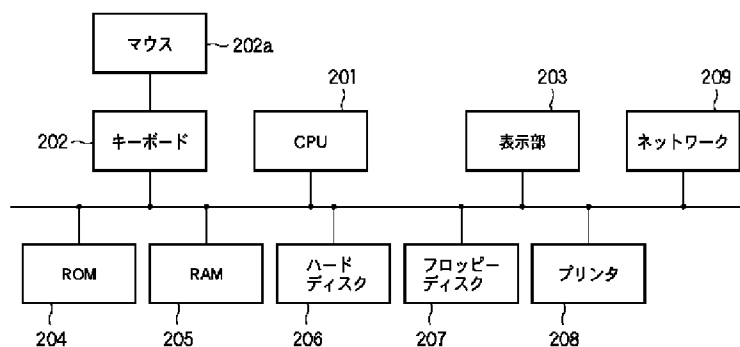
【図 1】



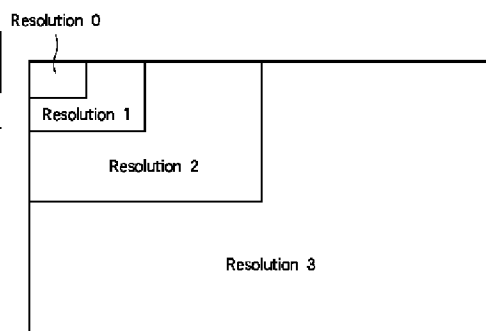
【図 3】



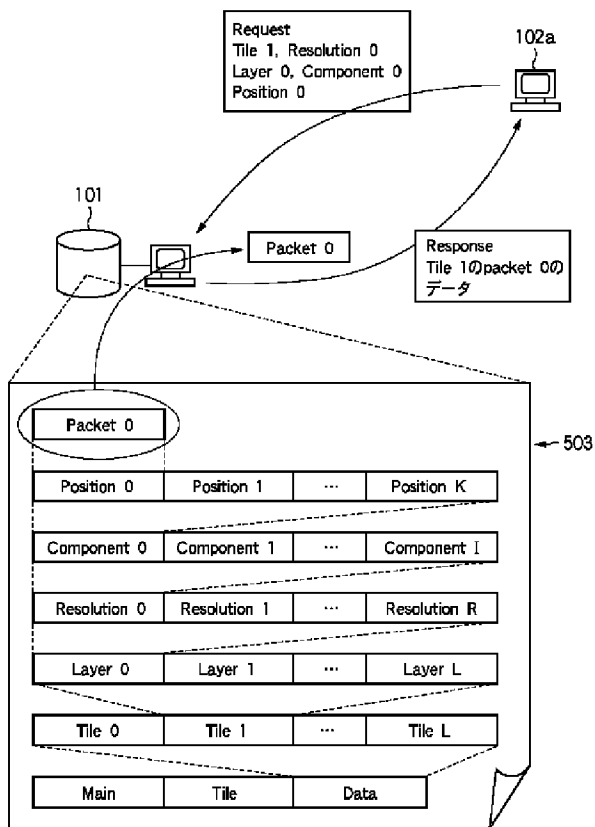
【図 2】



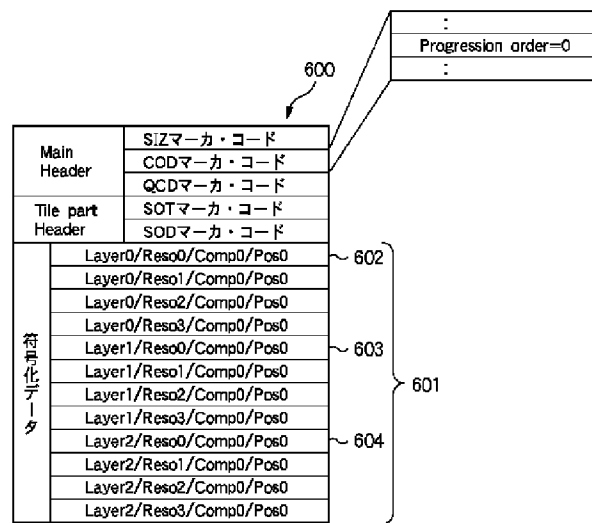
【図 4】



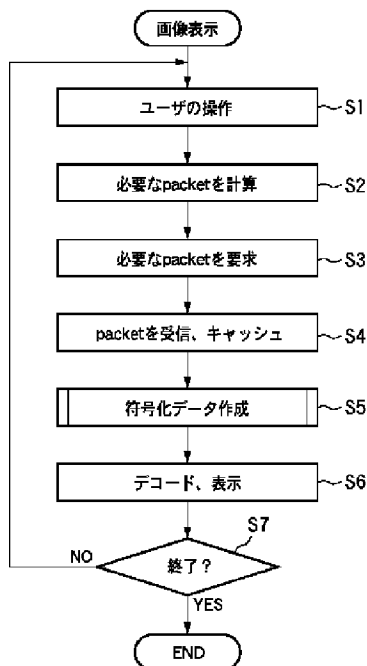
【図 5】



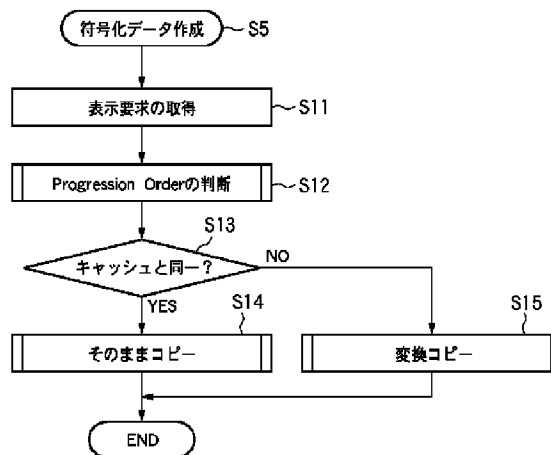
【図 6】



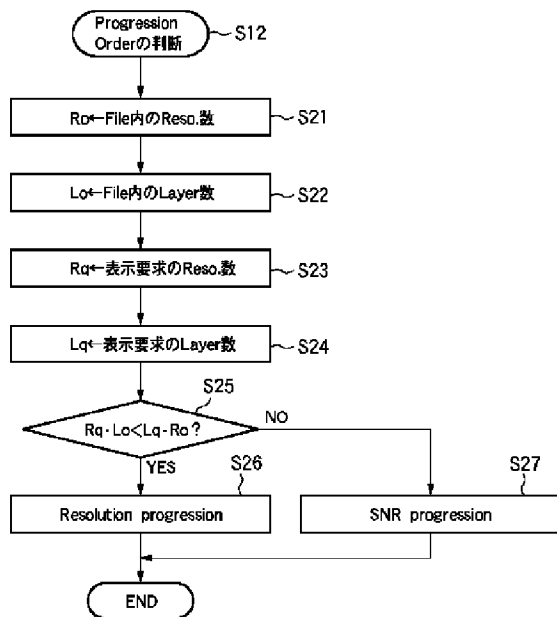
【図 7】



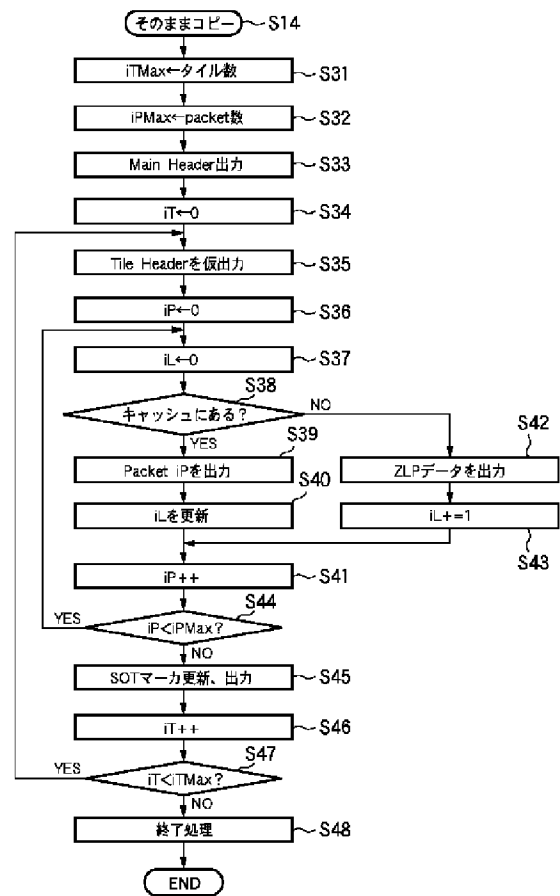
【図 8】



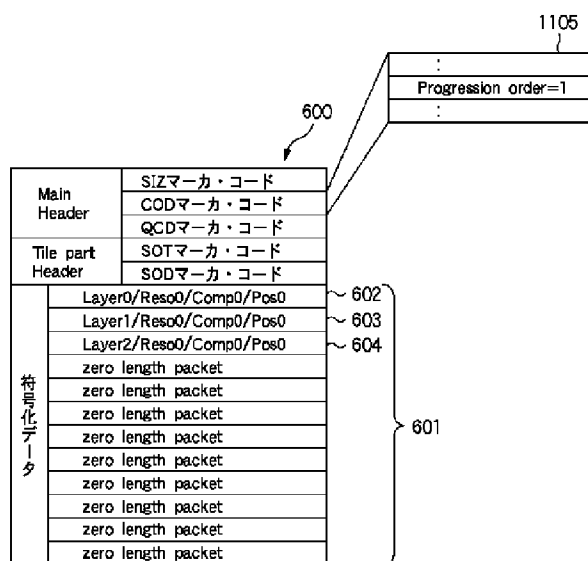
【図 9】



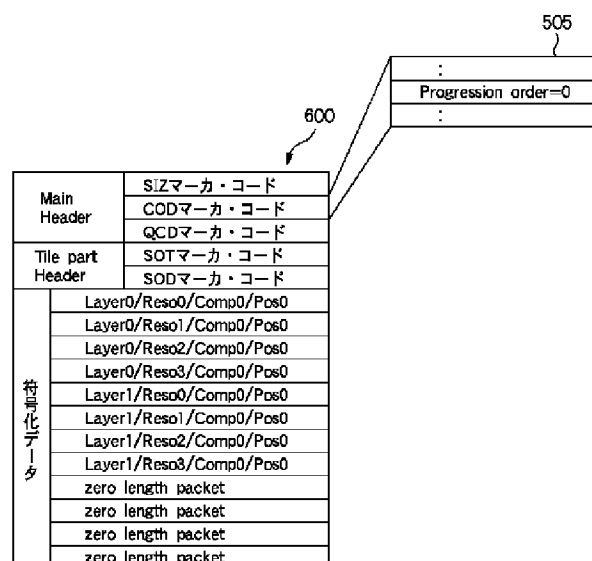
【図 10】



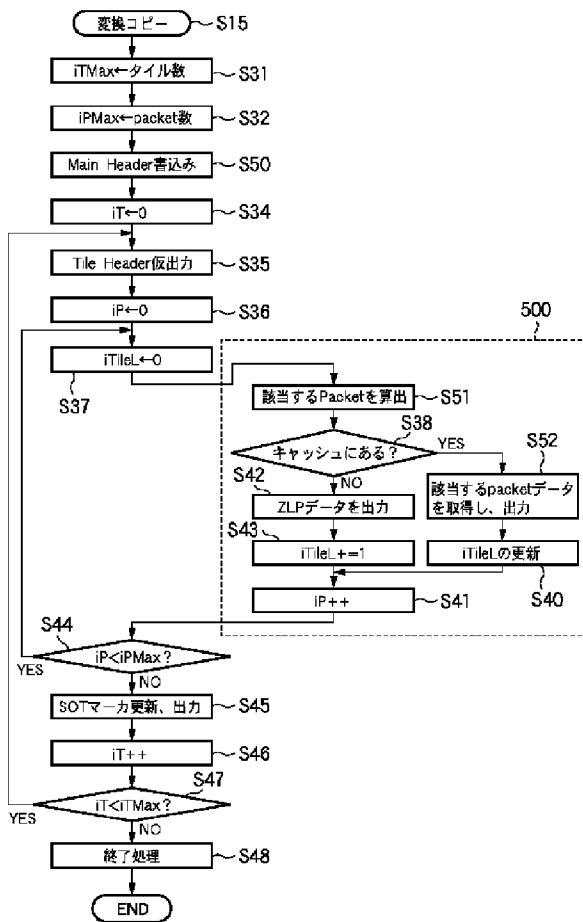
【図 12】



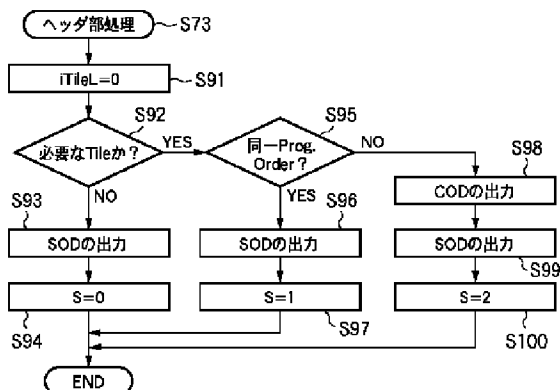
【図 13】



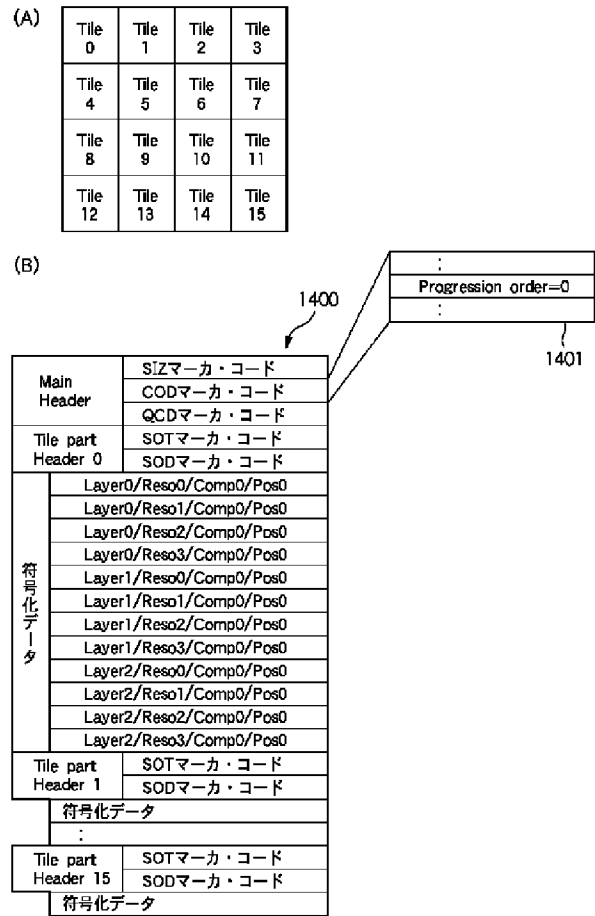
【図 1 1】



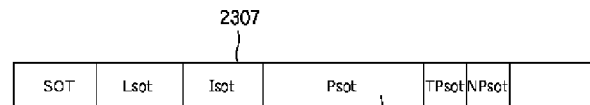
【図 1 7】



【図 1 4】



【図 2 3】

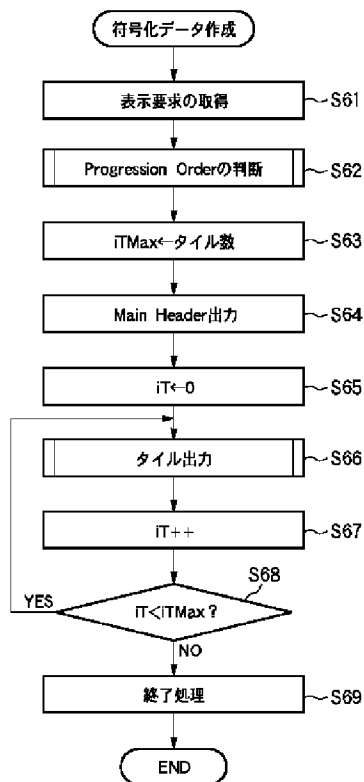


(A)

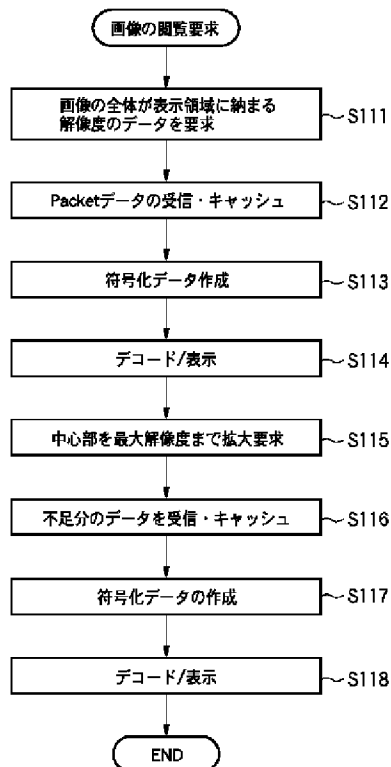
Parameter	Size(bits)	Values
SOT	16	0xFF90
Lsot	16	10
Isot	16	0
Psot	32	32
TPstot	8	0
NPstot	8	1
	16	0xFF94

(B)

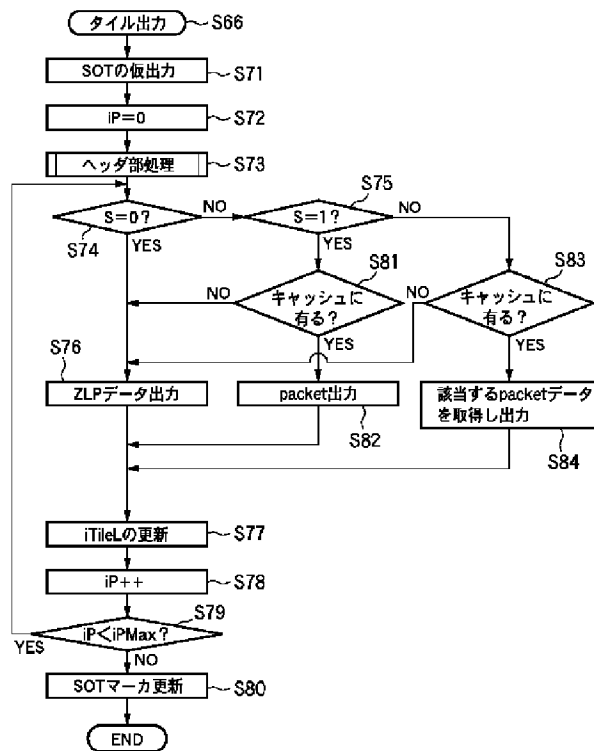
【図15】



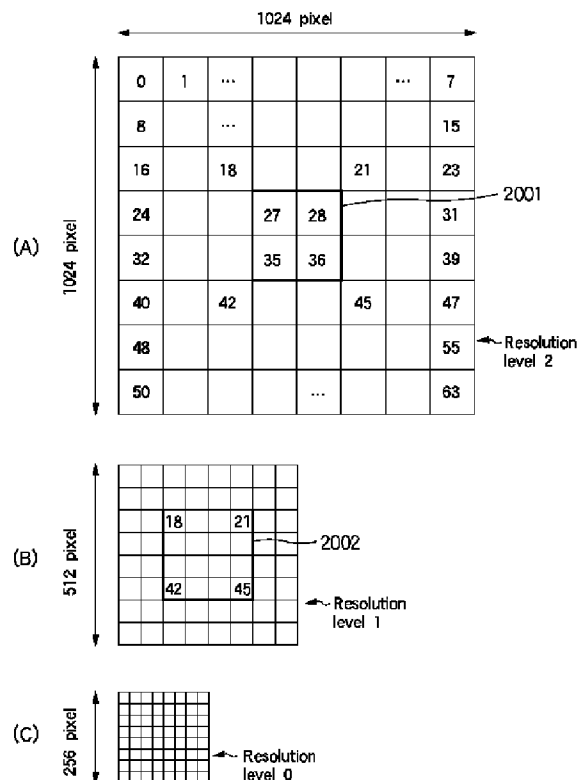
【図19】



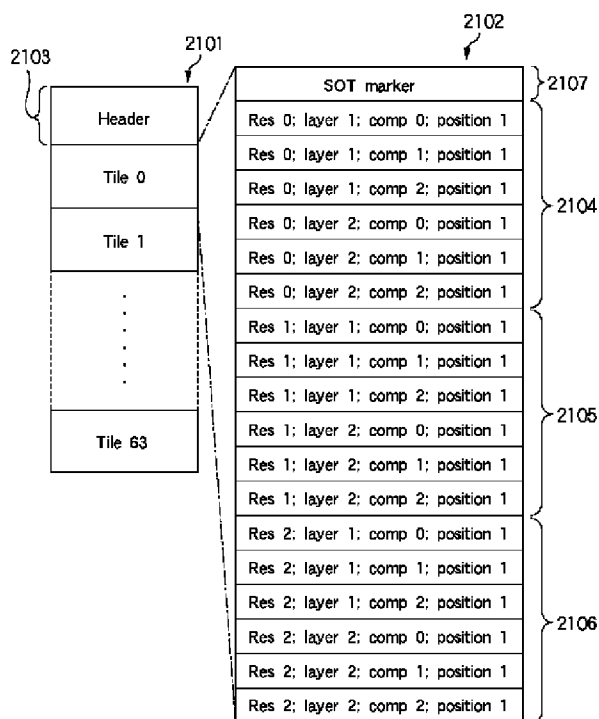
【図16】



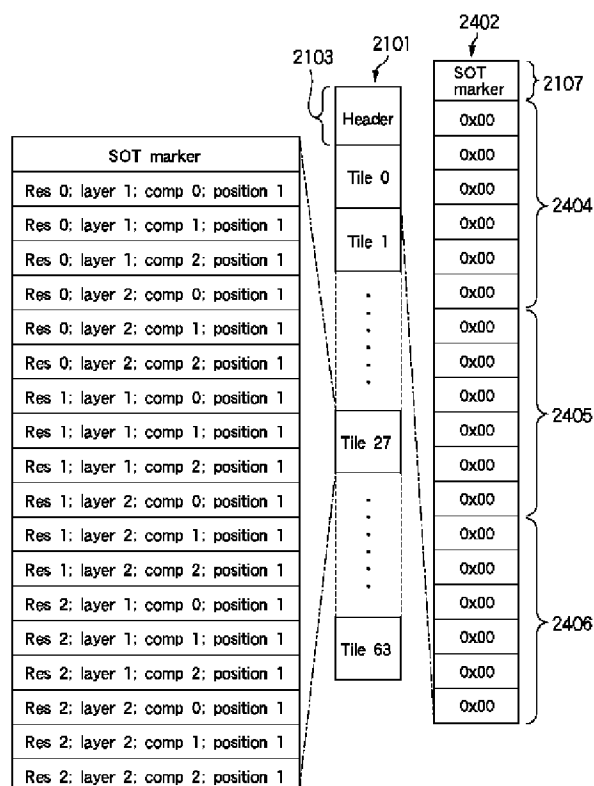
【図20】



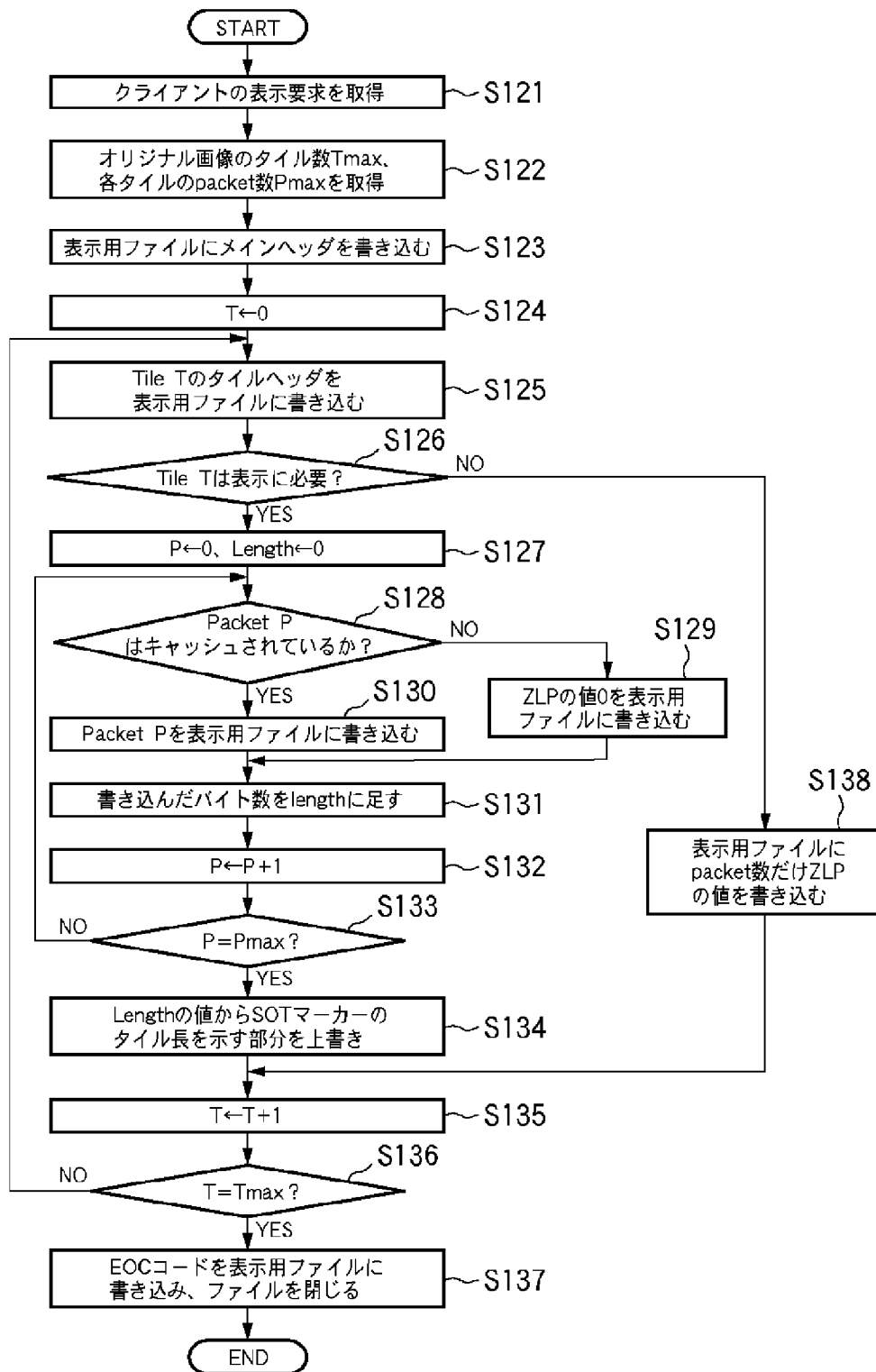
【图 2 1】



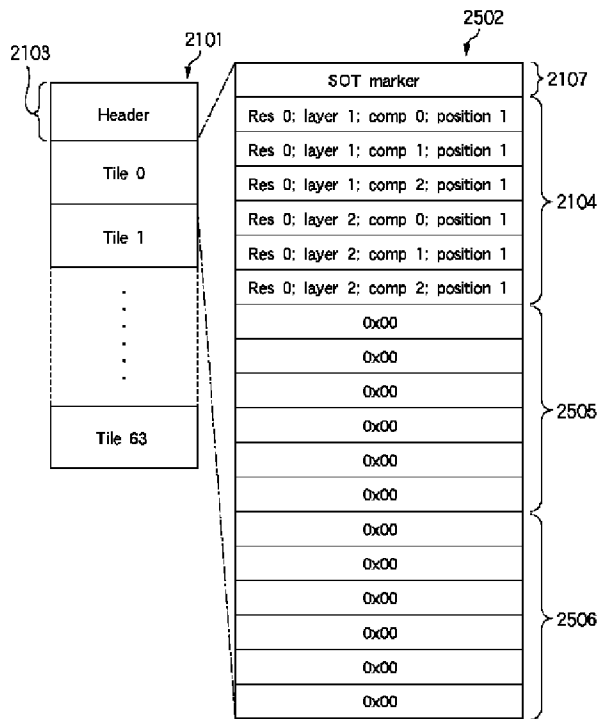
【图 24】



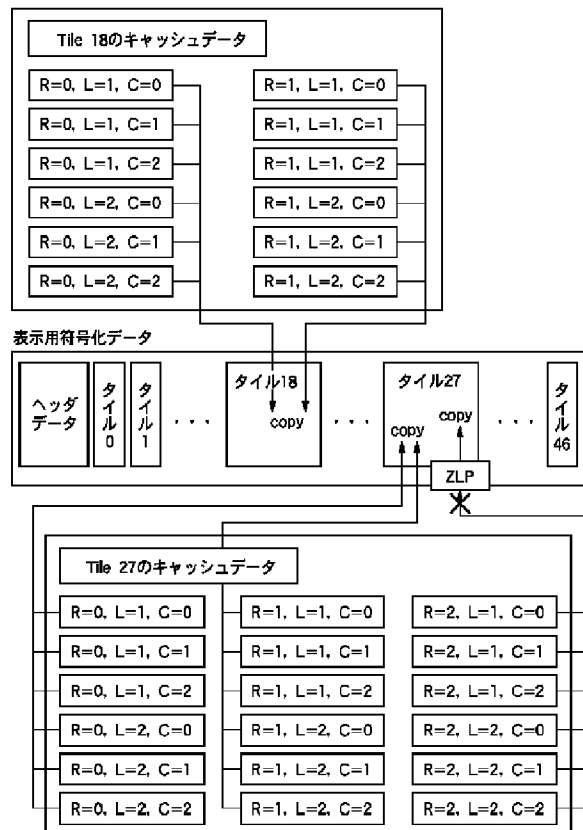
【図 2 2】



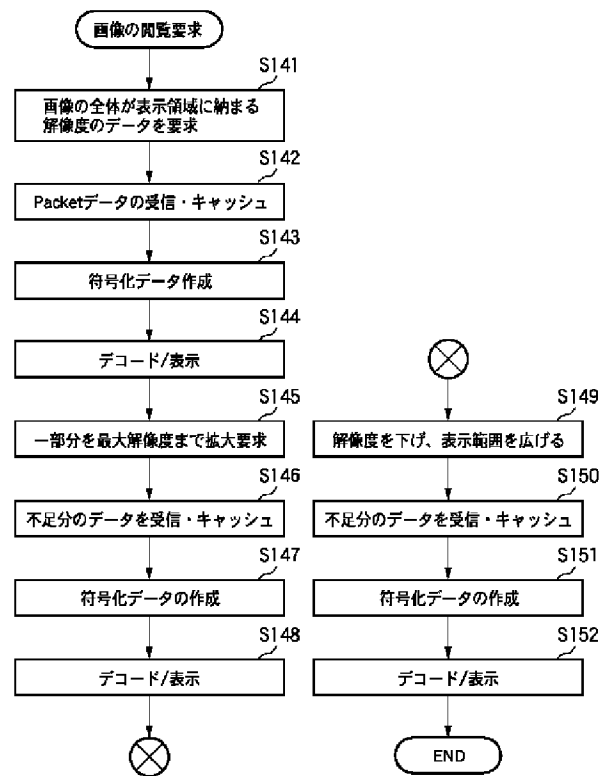
【図 2 5】



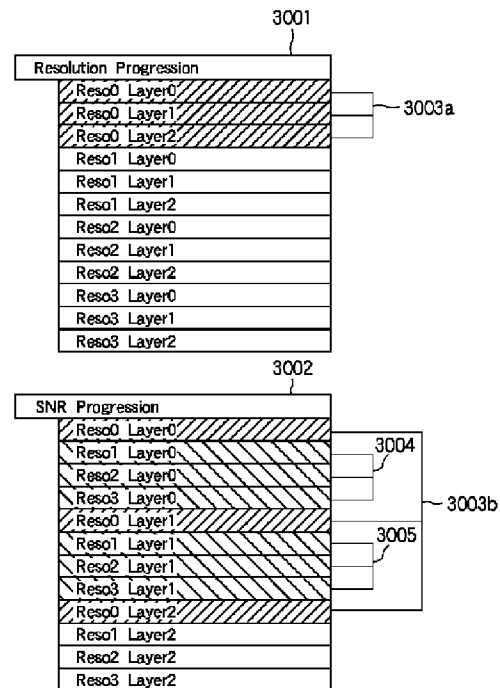
【図 2 9】



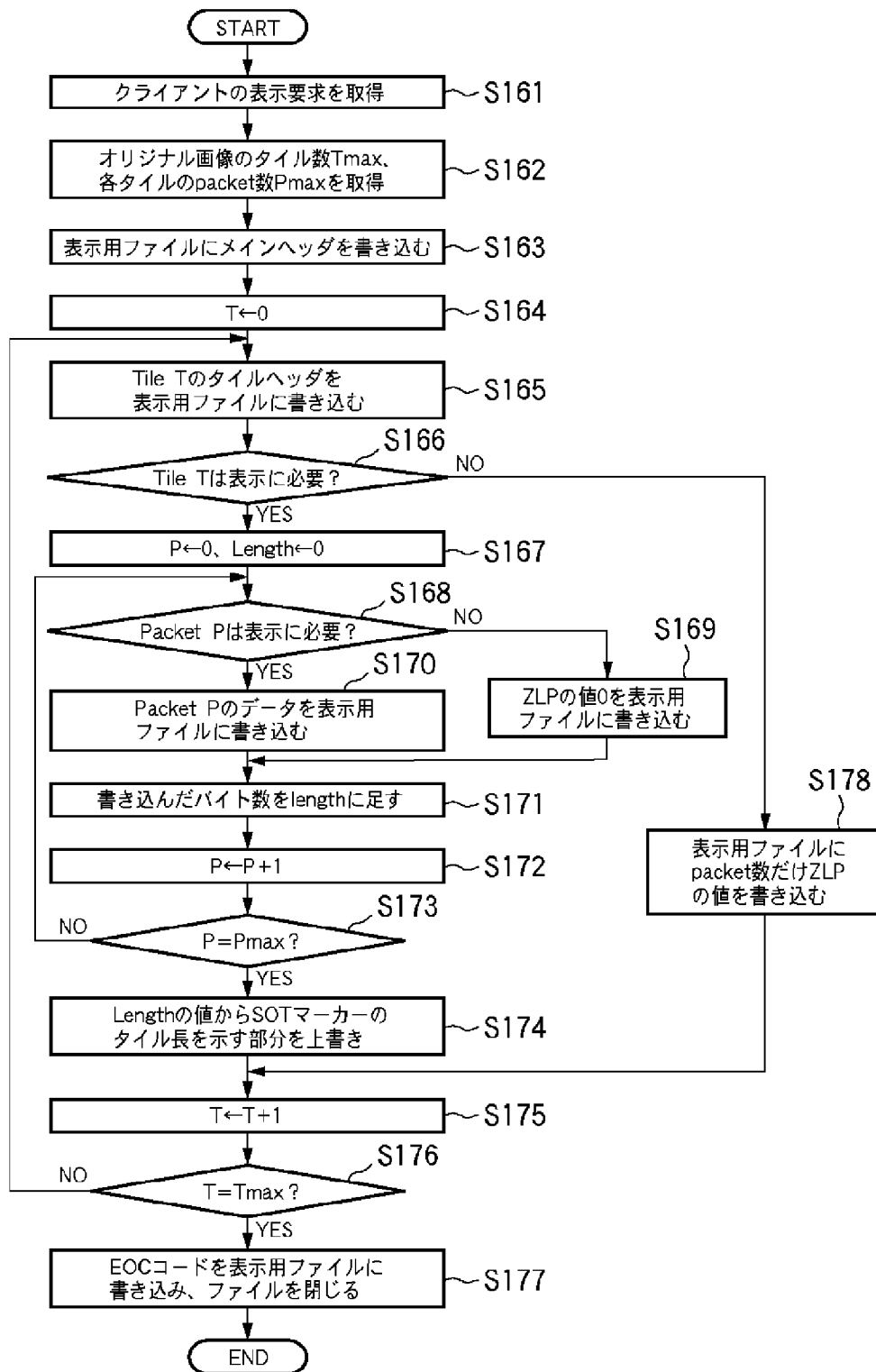
【図 2 6】



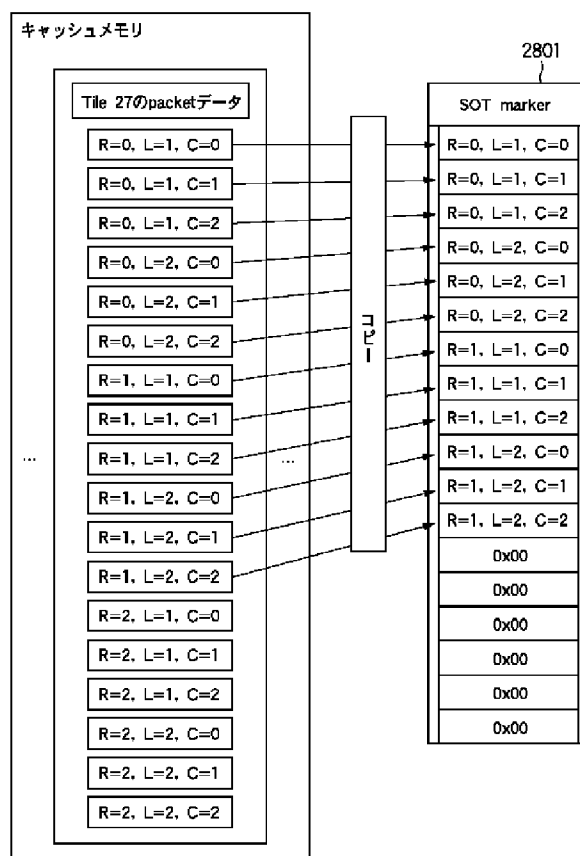
【図 3 0】



【図 27】



【図 28】



フロントページの続き

F ターム(参考) 5C059 KK11 MA00 MA24 MA32 MA34
 PP01 RA09 RB02 RE07 SS08
 SS20 TA00 TB00 TC01 TC27
 TC45 UA02
 5C078 BA57 CA34 CA39 EA01
 5J064 AA03 BA16 BB04 BC01 BC27

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-169216

(43)Date of publication of application : 13.06.2003

(51)Int.Cl. H04N 1/41
H03M 7/30
H04N 7/30

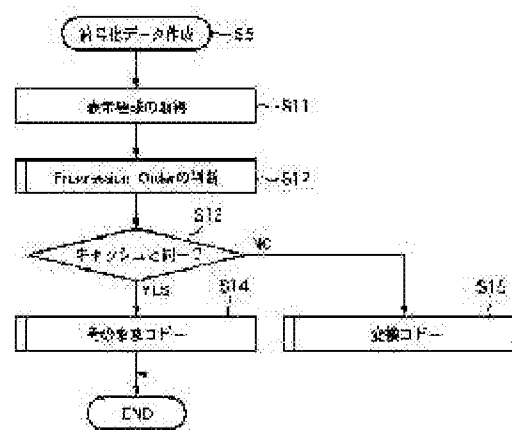
(21)Application number : 2001-364894 (71)Applicant : CANON INC
(22)Date of filing : 29.11.2001 (72)Inventor : ENOKIDA MIYUKI
ISHIKAWA CHIE

(54) METHOD AND APPARATUS FOR CREATING CODED DATA

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a method and apparatus for creating coded data that receives segmented coded-data, caches them and uses the cached coded-data to create coded-data meeting a display requirement.

SOLUTION: The method and apparatus for creating coded data of this invention includes steps of receiving the segmented coded-data to store them to a memory and judging the Progression Order on the basis of a display requirement of a user (S12), revising header information on the basis of the judgment, deciding whether or not the memory stores the corresponding coded data on the basis of the revised header information, reading the segmented coded-data decided to be stored in the memory, converting the read data into the Progression Order suitable for the display requirement and applying a ZLP (zero length packet) process to coded data not stored in the memory to create the coded data in compliance with the JPEG 2000.



LEGAL STATUS

[Date of request for examination] 11.06.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3768866

[Date of registration] 10.02.2006

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]